The Dissertation Committee for Himanshu Vijay Save
certifies that this is the approved version of the following dissertation:

Using Regularization for Error Reduction in GRACE Gravity Estimation

Committee:

Byron D. Tapley, Supervisor

Srinivas V. Bettadpur, Supervisor

David G. Hull

E. Glenn Lightsey

Clark R. Wilson

Using Regularization for Error Reduction in GRACE Gravity Estimation

by

Himanshu Vijay Save, B.Tech, M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2009

Dedicated to late Prof. Madhav N. Kulkarni, Department of Civil Engineering, Indian Institute of Technology Bombay.

Acknowledgments

I would like to thank my supervisor, Byron Tapley for his guidance, financial support and the privilege to participate in the scientific investigations of his vastly successful GRACE mission. The contributions of the other committee members are also greatly appreciated. I would like to thank Srinivas Bettadpur for serving as a mentor during my years as a graduate student at the Center for Space Research and throughout this research. It has been a privilege to work with him. His vast knowledge of the subject, patience and guidance helped shape my research to a large extent. I would like to thank Steve Poole, John Ries, Richard Eanes and Don Chambers for their expertise and help with various aspects of my study and Jennifer Bonin for being available to bounce off ideas and help with parts of the analysis. I would like to acknowledge the Center for Space Research staff who have made working there a great experience. I would like to acknowledge the Texas Advanced Computing Center (TACC) for providing access and support for the computational resources used in the development of this study. This research would not have been successful without their resources and support.

Lastly, I would like to recognize my wife, Shivani and my patents for their support and patience over the years. It is without exaggeration to say I would not have gotten this far without them.

Using Regularization for Error Reduction in GRACE Gravity Estimation

Publication No. ________________

Himanshu Vijay Save, Ph.D.
The University of Texas at Austin, 2009

Supervisors: Byron D. Tapley
Srinivas V. Bettadpur

The Gravity Recovery and Climate Experiment (GRACE) is a joint National Aeronautics and Space Administration / Deutsches Zentrum für Luft- und Raumfahrt (NASA/DLR) mission to map the time-variable and mean gravity field of the Earth, and was launched on March 17, 2002. The nature of the gravity field inverse problem amplifies the noise in the data that creeps into the mid and high degree and order harmonic coefficients of the earth's gravity fields for monthly variability, making the GRACE estimation problem ill-posed. These errors, due to the use of imperfect models and data noise, are manifested as peculiar errors in the gravity estimates as north-south striping in the monthly global maps of equivalent water heights.

In order to reduce these errors, this study develops a methodology based on Tikhonov regularization technique using the L-curve method in combination with orthogonal transformation method. L-curve is a popular aid for determining a suitable value of the regularization parameter when solving linear discrete ill-posed problems using Tikhonov regularization. However, the computational effort required to determine the L-curve can be prohibitive for a large scale problem like GRACE. This study implements a parameter-choice method, using Lanczos bidiagonalization that is a computationally inexpensive approximation to L-curve called L-ribbon. This method projects a large estimation problem on a problem of size of about two orders of magnitude smaller. Using the knowledge of the characteristics of the systematic errors in the GRACE solutions, this study designs a new regularization matrix that reduces the systematic errors without attenuating the signal. The regularization matrix provides a constraint on the geopotential coefficients as a function of its degree and order. The regularization algorithms are implemented in a parallel computing environment for this study. A five year time-series of the candidate regularized solutions show markedly reduced systematic errors without any reduction in the variability signal compared to the unconstrained solutions. The variability signals in the regularized series show good agreement with the hydrological models in the small and medium sized river basins and also show non-seasonal signals in the oceans without the need for post-processing.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Gravity missions - a historical background

Since the days of Isaac Newton, scientists have been interested in the Earth's gravity field, knowing that greater understanding of it would lead to a greater understanding of the structure of the planet. As geophysics evolved, it became apparent that a study of the Earth's gravity field could provide information on both the physical and mechanical properties of the tectonic plates, and give an idea of the processes going on deep below the plates. Ground based measurements were the basis for the understanding and the knowledge of gravity until the late 1960s. But there was absence of spatial coverage, lack of homogeneity and uniform reference which lead to the lack of accuracy of the gravity field which is vital for number of geophysical applications such as oceanography, hydrology and geodesy. The launch of the Soviet satellite "Sputnik" in the fall of 1957 started a new phase in the gravity field modeling. The use of satellite tracking data has helped improve the gravity model and there have been steady advancements in the techniques of gravity field determination and also the spatial resolution of the gravity field. Because, the Earth is non-spherical and has a heterogeneous mass distribution, its gravity field is non-uniform, thereby causing the motion of an orbiting satellite to be perturbed away from the relative two body orbital motion [55] given by,

$$\ddot{\vec{\mathbf{r}}} = -\frac{\mu}{r^2}\frac{\vec{\mathbf{r}}}{r} \tag{1.1}$$

where μ is the gravitational constant and r is the relative barycentric distance between the two bodies of point mass or uniform mass distribution. An estimated gravity solution is commonly a synonym for a set of parameters that are the coefficients of the orthogonal basis functions in a model that approximates the shape of the Earth's gravity field. These coefficients are often those of the Legendre polynomial and trigonometric functions in a spherical harmonic expansion that describes the three dimensional surface potential in the free space (zero density) above the Earth [29].

The model of the Earth's gravity field is derived mainly from two sources of data; terrestrial measurements and satellite tracking. Over the years, there have been several noteworthy gravity models, starting with Johns Hopkins University/Applied Physics Laboratory (JHU/APL) 1.0 which was developed in 1963. It was a degree and order 8 field and was followed by "Standard Earth 1" in 1966 developed by Smithsonian Astrophysical Observatory (SAO). Later, JHU/APL 4.5 was released in 1967 and contained resolution up to degree and order 15. The size of the gravity fields, as determined by the number of coefficients in the model, have increased during the following intervals based on the improvements in observation technologies and increase in computational power. Since 1972, NASA/GSFC (Goddard Space Flight

Center) developed a series of gravity models called “Goddard Earth Model” (GEM). These were derived from the sources of laser range data, Doppler range data, satellite radar altimetery and surface gravity data. GEM 10 released in 1982, was complete to degree and order 36, followed ten years later by GEM-T3 which was complete to degree and order 50. In a joint effort between NASA/GSFC and The University of Texas at Austin/Center for Space Research (UT/CSR), a follow-up gravity model, complete to degree and order 70, was released which was known as the Joint Gravity Model (JGM). UT/CSR has independently developed a number of gravity field models with the designation Texas Earth Gravity (TEG), using a variety of observations including satellite laser ranging (SLR) radar altimetery, Doppler range rate and surface gravity. TEG-1 with degree and order 36 and TEG-2 with degree and order 50 were published in 1988 and 1990 respectively. TEG-3 with degree and order 70, released in 1997, also included the first GPS data from the on board receiver of TOPEX/Poseidon and surface gravity from the Ohio State University (OSU) gravity model OSU91A.

Satellite tracking data has improved our gravity models over the years and the Global Positioning System (GPS) has contributed significantly to these efforts with its precise tracking of satellites such as TOPEX/Poseidon and CHAMP. The improvements in the models have been impressive, allowing for unprecedented accuracy in orbit determination of Earth orbiting satellites. But, these models had little data from satellites with orbits less than 800 km altitudes and also had noticeable gaps in gravity information in certain inclination bands due to lack of data from satellites in those inclination bands. The variations in the gravity field of the earth occur with time scales between a few hours and decades due to the fluid and atmosphere dynamics in, on and above the Earths surface. The signals of these time variations of the gravity field were missing in the gravity models derived from the satellites before the year 2000. A satellite mission flying at low altitude with satellite to satellite tracking was proposed as one method for obtaining a more accurate gravity field model. There have been a variety of studies concerning the achievement possible using a low-low satellite to satellite tracking (SST) mission since its initial concept was first proposed by Wolf [59].

The need for a dedicated gravity mission using two low altitude satellites led to NASA proposing the Geopotential Research Mission (GRM) in the early 80s. GRM was proposed to globally determine high precision gravitational and magnetic fields of Earth using two nearly identical satellites in a nearly circular, 160 km altitude orbit [30]. This mission was to use a microwave ranging system to measure the range-rate between the two satellites, while thruster firings were used to control/correct for the effects due to the non-gravitational forces such as drag and radiation pressure [30] [38]. Later in the 1990's two new low-low satellite to satellite missions were proposed. The first mission was the Gravity and Magnetic Earth Surveyor (GAMES) mission [56], which was to use two satellites, one active and one passive. The active satellite was to be equipped with a laser to track the passive satellite and would in turn be tracked via a GPS receiver. The primary satellite was to carry an on board accelerometer to measure the non gravitational forces. The second mission proposed was the Gravity Recovery and Climate Experiment (GRACE).

1.2 The GRACE mission

In May of 1997, NASA announced its plan to fly the GRACE mission and a dedicated space borne mission to map the Earth's gravity field was launched in March 2002. GRACE mission is jointly implemented by NASA and the German Aerospace Center, DLR, under the NASA Earth System Science Pathfinder (ESSP) Program. Dr. Byron Tapley of UT/CSR is the Principal Investigator (PI), and Dr. Christoph Reigber of GFZ is the Co-Principal Investigator (Co-PI) of the mission. Project management and systems engineering activities are carried out by the Jet Propulsion Laboratory (JPL). The ground based tracking and operations of GRACE is handled by the German Space Operations Center (GSOC) with the observation data collected being processed by JPL, UT/CSR and GeoForschungsZentrum (GFZ).

The mission consists of two co-orbiting satellites in a near polar and near circular orbits separated by an along track distance around 220 km. The satellites were injected in the orbit at the altitude of about 500 km and will descend to a final altitude of approximately 300 km before re-entry into the Earth's atmosphere. The primary mission of GRACE is to obtain accurate global models for the mean and time variable portions of the Earth's gravity field. As the satellites move over an area with an increased mass concentration, they are first pulled forward as they approach, then backwards as they pass. The 220 km separation means that the two satellites accelerate and decelerate at different times, leading to an oscillation in the inter-satellite range. This oscillation is a measure of the gravity due to the mass concentration. Figure (1.1) depicts the basic geometry of the mission.

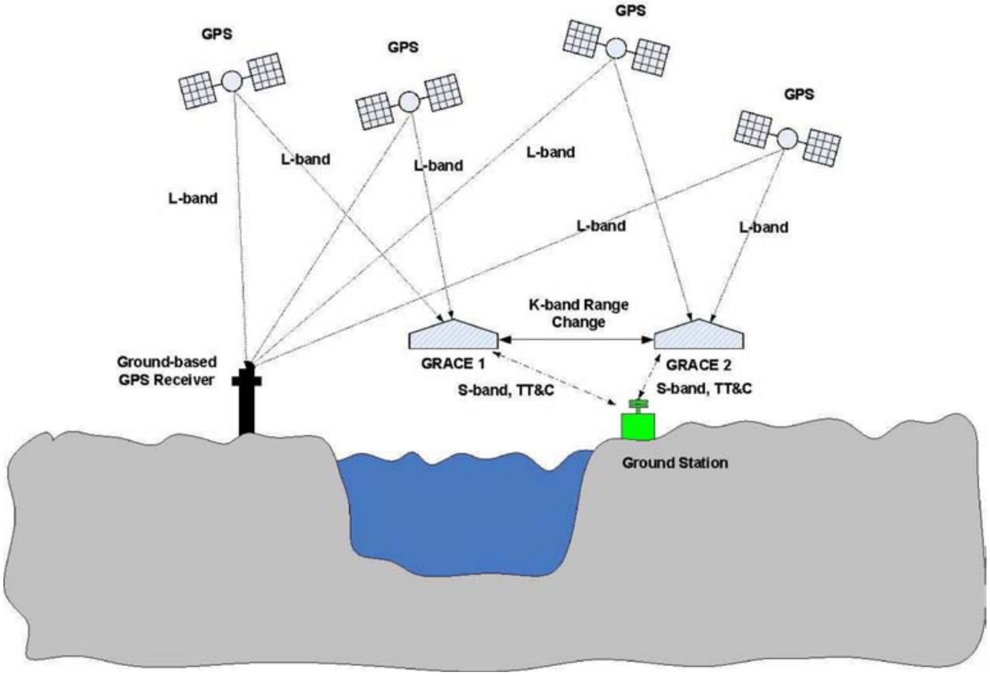


Figure 1.1: Geometry of the GRACE mission [32]

The fundamental measurement for gravity recovery is the inter-satellite range change, which is provided by a K-Band Ranging system (KBR) that determines the change in the range to a precision of a few microns. In order to measure this range and range-rate precisely, the GRACE satellites use a dual one-way microwave ranging system [14]. Identical transmission and reception sub-systems are present on each satellite and transmit the carrier signal to the other satellite. The received signals of both the satellites are down-linked to the ground stations and then combined. In order to remove the ionospheric delay, both K and Ka bands are used, thus providing a precision of the inter-satellite range rate of 0.2 to 0.4 $\mu m/s$ RMS noise. To maximize the data quality, the satellite attitudes are maintained so that the K-Band antennas are continuously pointed at each other. This requires precise attitude control and a one degree pitch on both satellites at the nominal 220 km separation. The attitude is determined with two star cameras on each satellite, and controlled by cold gas thrusters and magnetorquer rods. Orbit determination utilizes the GPS system, which also assists with the inter-satellite range measurement. Because gravity is not the only force affecting the satellites, the effects of drag and other surface forces are measured with the SuperSTAR accelerometer [53]. Figure (1.2) shows the layout of these systems in the satellites.

The determination of gravity from the satellite range changes is an iterative process of improving an existing gravity field model. The range changes measured by GRACE are due to variations in the true gravity, plus the effect of other forces on each of the satellites like drag, radiation pressure and thruster activations. The expected range changes are predicted using the current best gravity model and measurements of the non gravitational forces from the SuperSTAR accelerometer. In principle, if the non-gravitational forces from the accelerometer are accurately measured on the satellite, then the residuals between the measured and the predicted range changes will only be due to the

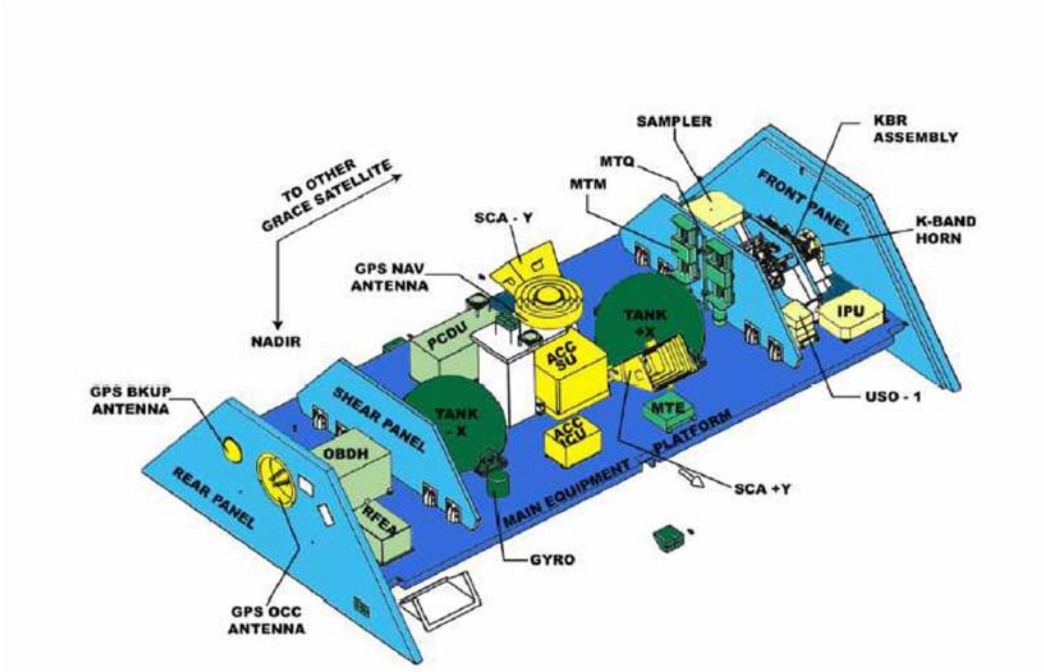


Figure 1.2: Layout of the GRACE instruments on the satellite [14]

omissions and errors in the initial gravity field model. These residuals can be used for differential corrections of the gravity model.

GRACE has increased the accuracy of the estimate of the Earth's gravitational potential model, as defined by the spherical harmonics expansion coefficients, to unprecedented levels [50]. GRACE measures the gravity in two ways, the long-term mean field and the short-term temporal variability.

1.3 Study objectives and outline

This study looks at regularization of the GRACE gravity estimation to reduce the systematic errors introduced by the ill-posedness of the problem. This is a consequence of the observability problem which is discussed in the chapter 2. The two main reasons for the ill-posedness of the GRACE gravity estimation problem are *downward continuation* and *ground coverage*.

- Downward Continuation

  According to Newton's inverse-square law for the gravitational force, gravity fields decay with distance away from their sources. This decay is dependent on wavelength, so that for any given distance away from the source, short-wavelength anomalies are attenuated more strongly than long-wavelength anomalies. The amount of attenuation depends on both the wavelength of the anomaly and the distance from the source [10]. The high degree and order harmonics have very slight effect on the satellite's orbit because these parameters are attenuated rapidly with increasing altitude. Hence, the satellite tracking data will observe only very small effects in these high degree and order co-efficient and is vulnerable to unwanted signals from noise and other undesirable sources. The attenuation with altitude has important consequences for satellite gravity missions. A satellite orbiting the Earth at some altitude h above the surface experiences an attenuated version of the gravity field on the surface of the Earth. The measurements made by that satellite must be downward continued (amplified by the reciprocals of the attenuation factors) to produce a gravity field on the surface, and in doing so, measurement errors are amplified [10]. This inability of the observations to capture the short-wavelength anomalies give rise to the observability problem.

- Ground Coverage

  The quality of the temporal variability obtained by GRACE depends on how good the ground coverage is for the month. If we don't have a complete ground coverage, we try to estimate the parameters that have not been observed by the observations. This adds to the observability problem.

If we are not able to remove these observability problems by other means, we are forced to solve an ill-conditioned system of equations. This introduces systematic errors in the geo-potential estimate and one of the manifestation of these errors are the unphysical north-south striping in the variability maps obtained from GRACE. This study looks at an alternative method to estimate the gravity parameters by using regularization (or stabilization). Regularization process makes the ill-posed problem more well-conditioned by adding *pseudo-information* to the problem. Chapter 2 begins with an introduction to the observability problem and then describes the different methods available for tackling the observability problem during the solution process. Chapter 2 also describes the motivation for using the L-ribbon method for this study and discusses techniques and algorithms. Regularization of a large estimation problem like that of GRACE is usually computationally very expensive. It also describes the parallelization of the Lanczos bidiagonalization algorithm, which is the critical component in making the regularization of a large problem like GRACE, computationally inexpensive by implementing the L-ribbon approach to regularize the GRACE solutions [5]. Chapter 3 describes the choices made with respect to the regularization of the GRACE problem and deals with the deign of the regularization matrix used in this study. The quality of the candidate regularized solutions generated are evaluated for the error and signal content in chapter 4. Chapter 5 concludes the study with recommendations for the optimal regularization scheme and future studies.

Chapter 2

Regularization

2.1 Linearized problem

Consider an overdetermined system of equations given by

$$y = Hx + \epsilon \tag{2.1}$$

where y is an $m \times 1$ observation vector, x is an $n \times 1$ state vector to be estimated, ϵ is an $m \times 1$ vector of observation errors and H is an $m \times n$ mapping matrix with n being the number of state variables to be estimated and m being the total number of observation. Such system with $m \geq n$ are usually solved using least square analysis. The least squares criterion provides us with conditions on the m observation errors that allow a solution for the n state variables, x.

The estimate $\hat{\mathrm{x}}$ in equation (2.1) is obtained in the normal equations approach by minimizing the square of the errors as in the following equation.

$$J = min\{\|Hx - y\|^2\} \tag{2.2}$$

The details about the formulation of such linear system for GRACE estimation problem is given in the Appendix A which also outlines the parameter space and the observation datasets used for the estimation. It also outlines the weighted least squares formulation with an optimal weighting scheme implemented by GRACE.

2.2 Ill-conditioned systems

Ill-conditioned problem is defined as a problem that does not have a unique solution or is not a continuous function of the data (i.e., if an arbitrary small perturbation of the data can cause an arbitrary large perturbation of the solution) [24]. A system of linear equations is considered to be ill-conditioned when it does not contain enough information to determine all of the parameters that one is attempting to estimate with confidence. If such a system is not observing all of the intended parameters, then it cannot resolve every one of them unambiguously during the solution. The discussion of the ill-conditioned system requires the knowledge of singular value decomposition (SVD) (section 2.3) of the coefficient matrix H. In particular the conditioning number of H which is defined as the ratio between the largest and the smallest singular values of H. The numerical treatment of systems of equations with an ill-conditioned coefficient matrix depends on the type of ill-conditioning of H. There are two important classes of problems to consider with respect to ill-conditioned problems, and many practical problems belong to one of these two classes.

1. Rank deficient problems

   These are characterized by the matrix H having a cluster of small singular values and there is a well defined gap between the large and small singular values. This implies that one or more rows and columns of the matrix H are nearly linear combinations of some other or all of the remaining rows or columns. The matrix H thus contains almost redundant information. The key to the numerical treatment of such problems is to extract the linearly independent information in H, to arrive to another problem with a well-conditioned matrix.

2. Discrete ill-posed problems

   These problems have all the singular values of H as well as the singular value decomposition (SVD) components of the solution, on an average, decay gradually to zero. Since there is no gap in the singular value spectrum, there is no notion of a numerical rank for these matrices. For solving these problems the goal usually is to find a balance between the residual norm and the size (norm) of the solution that matches the errors in the data as well as one's expectations to the computed solutions. The GRACE estimation process is a discrete ill-posed problem, where the singular values of H decay gradually while giving a large conditioning number which is illustrated in the Figure 2.1 [26]. The largest singular value in the Figure 2.1 is 1.04×10^7 and the smallest singular value is 8.29×10^1 which gives a conditioning number of 1.25×10^5 as illustrated by Hinga [26]. Though the smallest singular value is not close to zero, it gradually decreases and gives a very high conditioning number making the problem ill-posed. The method we employ to solve the ill-posed

GRACE gravity estimation problem will find the balance between the residual norm and the solution norm which strikes a balance between the amount of signal and noise in the solution.

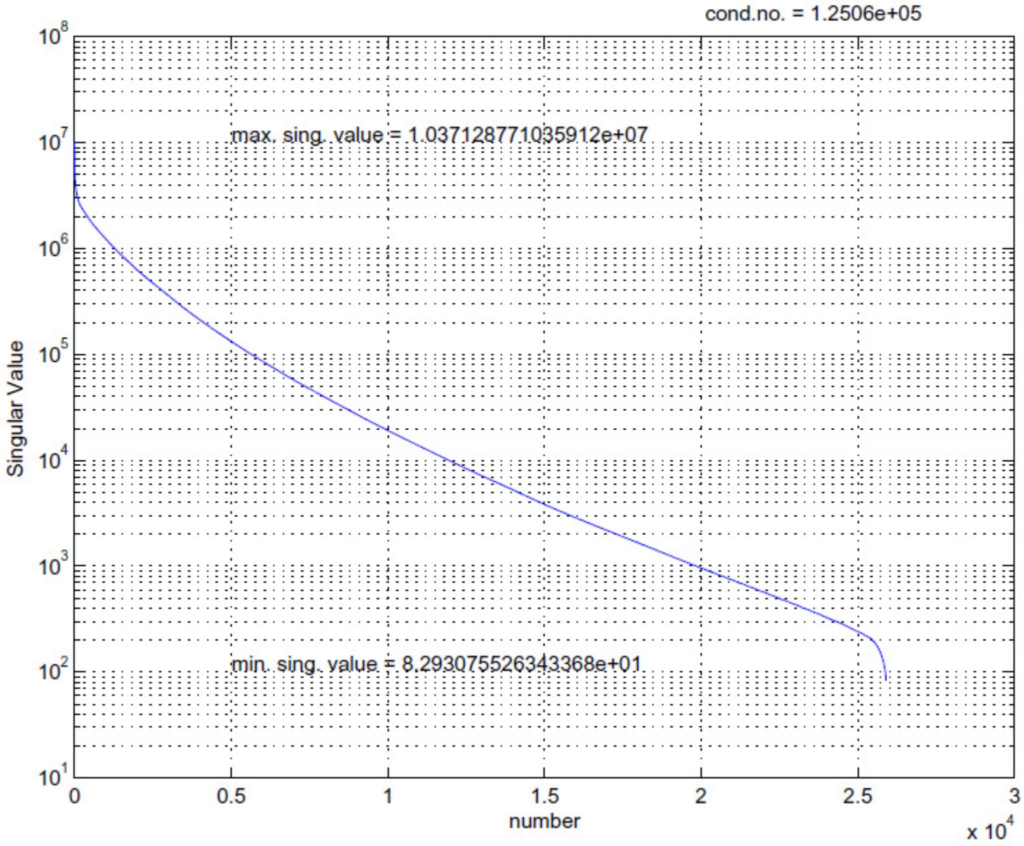


Figure 2.1: Singular values of a sample GRACE problem [26]

Both these classes of problems are under-determined because of the large condition number of H. Both the classes of problems can be solved using a technique called regularization which stabilizes the problem using some extra *pseudo-information* that is usually known about the problem. This chapter discusses the numerical methods that are suited for regularization of GRACE estimation problem. Both the classes of problems have a strong relation between the amount of extracted linearly independent information and the norm of the solution and the corresponding residual. Nevertheless, it is often advantageous to keep in mind the basic difference between the two problem classes, namely, a gap in the singular value spectrum verses an overall decay. Some of the methods used to solve these problems are described in this chapter.

In order to discuss the observability problems of an ill-posed linear system of equations, consider a linear system of equation given by,

$$Ax = b \tag{2.3}$$

where A is square non-singular and b is non zero and there is a unique solution x, which is non zero. Thus the solution to the equation (2.3) is given by

$$x = A^{-1}b \tag{2.4}$$

Now suppose we add a small perturbation δb to b. This system of equations also has a unique solution $\hat{x}$ which should not be too far from x. Let δx denote this difference between x and $\hat{x}$ so that $\hat{x} = x + \delta x$. If we introduced $\|\cdot\|$ as a vector norm, we can define the relative sizes of δx with respect to (wrt) x and δb wrt b as $\frac{\|\delta b\|}{\|b\|}$ and $\frac{\|\delta x\|}{\|x\|}$ respectively. It is desired that if $\frac{\|\delta b\|}{\|b\|}$ is small, then $\frac{\|\delta x\|}{\|x\|}$ is also small. Substituting the perturbation equations into equation (2.3) we get, $A(x + \delta x) = (b + \delta b)$, which yields $A\delta x = \delta b$ or $\delta x = A^{-1}\delta b$.

Thus, by invoking the induced matrix norm, since A is assumed to be non-singular, we get [58],

$$\|\delta x\| \leq \|A^{-1}\|\|\delta b\| \tag{2.5}$$

Similarly, $Ax = b$ or $b = Ax$ yields, $\|b\| \leq \|A\|\|x\|$

$$\frac{1}{\|x\|} \leq \|A\|\frac{1}{\|b\|} \tag{2.6}$$

Multiplying equations (2.5) and (2.6) we arrive at,

$$\frac{\|\delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\|\frac{\|\delta b\|}{\|b\|} \tag{2.7}$$

Equation (2.7) provides the bounds for $\frac{\|\delta x\|}{\|x\|}$ in terms of $\frac{\|\delta b\|}{\|b\|}$. The factor $\|A\|\|A^{-1}\|$ reflects the condition number of A. From this equation it can be seen that, if $\|A\|\|A^{-1}\|$ or the condition number of A is small then small values of $\frac{\|\delta b\|}{\|b\|}$ yields small values of $\frac{\|\delta x\|}{\|x\|}$ which is acceptable. But if the condition number of A is big then small values of $\frac{\|\delta b\|}{\|b\|}$ might cause a large change in $\frac{\|\delta x\|}{\|x\|}$. In other words, the system with high condition number for A, is extremely sensitive to perturbations in b and may cause very large perturbations in x which leads to the ill-posed problem. The problem of solving any ill-conditioned system requires the user to know what kind of ill-conditioning to expect and to know which numerical regularization method should be used to treat the problem efficiently and reliably on the computer.

2.3 Singular value decomposition (SVD)

The singular value decomposition can be used to find the rank of a given matrix in order to determine if it is invertible in the ordinary sense. If the matrix is rank deficient, which implies an ill-conditioned system, then its inversion is possible by using fewer singular values than the number of parameters that are being estimated. Removing the singular values to allow for a stable pseudo-inversion does not affect the parameter space. A full set of parameters come out of the solution process regardless of the number of singular values removed.

Let $H = U\Sigma V^T$ denote the SVD of H, where the u_i of U and v_i of V contain, respectively, the left and right singular vectors of H, and $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is the diagonal matrix containing the singular values of H, with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. Then the least square solution is given by using only the non-zero singular values in reconstructing the solution.

$$x_{LS} = \sum_{i=1}^{rank(A)} \frac{u_i^T y}{\sigma_i} v_i \tag{2.8}$$

In practice H is never exactly rank deficient, but instead numerically rank deficient; i.e. it has one or more small but nonzero singular values. In such cases, when one computes the least square solution, the norm of the solution can be very large due to very small singular values (σ_i).

$$\|x_{LS}\| = \sum_{i=1}^{n} \left(\frac{u_i^T y}{\sigma_i}\right)^2 \tag{2.9}$$

One of the approaches to solve these problems is to replace the matrix H with a matrix that is close to H and is mathematically rank deficient. The standard choice is to use a rank-k matrix A_k defined as

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T \tag{2.10}$$

where we replace the small non zero singular values $\sigma_{k+1}, \ldots, \sigma_n$ with exact zeros. We then use equation (2.8) to solve for the new rank-deficient problem. The number of small singular values removed determines the quality of the solution. This method is commonly referred to as truncated singular value decomposition (TSVD). There are multiple recommendations for the optimal choice of k in the literature e.g. [24]. The treatment of the GRACE gravity problem using SVD has been studied in detail by Hinga [26].

2.4 Tikhonov regularization

Let

$$y = Hx + e \tag{2.11}$$

where H is an $m \times n$ matrix with $m \geq n$, and y and e are vectors of size m, and x is a n-vector. The matrix H and y are known from the equations described in appendix A and e is assumed to be random noise.

The estimate x̂ in equation (2.11) is obtained in the normal equations approach by minimizing the following equation.

$$J = min\{\|Hx - y\|^2\} \tag{2.12}$$

If the matrix H is ill-conditioned and if neither the vector **e** nor its norm is known then the least square solution obtained from equation (2.12) is not a good approximation of x̂ and is contaminated with noise. Tikhonov regularization, also known as ridge-regression, is a popular approach used to get a more meaningful solution. Instead of minimizing the error in equation (2.12), one determines the solution of the minimization problem that includes some *pseudo-information* about the system of equations or some knowledge about the expected structure of the solution (for eg. the second derivative of the parameters or its spread.) For this approach, the solution x is selected to minimize the following expression.

$$J = min\{\|Hx - y\|^2 + \mu\|Mx\|^2\} \tag{2.13}$$

where M is referred to as the regularization operator and $\mu \geq 0$ as the regularization parameter. This equation can be converted to a generalized form with $M = I$ as follows

$$J = min\{\|\bar{H}\bar{x} - y\|^2 + \mu\|\bar{x}\|^2\} \tag{2.14}$$

where $\bar{H} = HM^{-1}$ and $\bar{x} = Mx$. It is suggested that the choice of M be such that it should be invertible but could be near singular [5]. The restriction that the M be invertible is applied to simplify the numerical methods significantly.

Equation (2.12) has a unique solution given by

$$\begin{aligned}\bar{x}_\mu &= (\bar{H}^T\bar{H} + \mu I)^{-1}\bar{H}^T y \\ &\mathit{or} \\ x_\mu &= (H^T H + \mu M^T M)^{-1} H^T y \end{aligned} \tag{2.15}$$

If $\mu = 0$, then the solution obtained x_μ is equivalent to the least squares solution. The estimation problem is thus stabilized by adding *pseudo-information* with $\mu M^T M$.

2.5 Parameter choice

The choice of the parameter μ is crucial, and many methods have been proposed in the literature for this purpose.

2.5.1 Morozov's Discrepancy Principle

In this method, the value of μ is chosen such that the norm of the residual $(y - \bar{H}\bar{x})$ or $(y - Hx)$ equals the norm of the error term [19].

$$\left\| y - \bar{H}(\bar{H}^T\bar{H} + \mu I)^{-1}\bar{H}^T y \right\| = \|e\| \tag{2.16}$$

For $\mu > 0$ the identity

$$I - \bar{H}(\bar{H}^T\bar{H} + \mu I)^{-1}\bar{H}^T = \mu(\bar{H}\bar{H}^T + \mu I)^{-1} \tag{2.17}$$

holds. Thus the the equation (2.16) can also be written as,

$$\phi_M(\mu) := \mu^2 y^T(\bar{H}\bar{H}^T + \mu I)^{-2} y - \|e\| = 0 \tag{2.18}$$

Because,

$$\phi'_M(\mu) := 2\mu y^T \bar{H}(\bar{H}\bar{H}^T + \mu I)^{-3}\bar{A}^T y \tag{2.19}$$

the function ϕ'_M is strictly increasing for $\mu > 0$, and the equation (2.18) has a unique solution. The drawback of this method is that we need to know the norm of the error vector in order to use this method. In most practical applications there is no information about the norm of the observation error and hence this approach cannot be used.

2.5.2 The Quasi-Optimality Criterion

The quasi-optimality condition is based on the assumption that the function $\|x^{exact} - x_\mu\|$, where x^{exact} is the exact solution of the unperturbed problem, contains a minimum at the optimum μ.

$$\|x^{exact} - x_\mu\| \sim (y^T(\bar{H}\bar{H}^T + \mu I)^{-4}\bar{H}\bar{H}^T y)^{\frac{1}{2}}$$

The minimization of this error estimate is given by

$$\begin{aligned} \phi_Q(\mu) &= \left\| \mu^2 \frac{dx_\mu}{d(\mu^2)} \right\| \\ &= \frac{1}{2}\left\| \mu \frac{dx_\mu}{d\mu} \right\| \end{aligned}$$

The quasi-optimality criterion [19] thus determines $\mu > 0$ such that μ is a global minimizer of

$$\phi_Q(\mu) := min\left\{\mu^2 y^T \bar{H}(\bar{H}^T\bar{H} + \mu I)^{-4}\bar{H}^T y\right\} \tag{2.20}$$

It should be noted that we have $\phi_Q(\mu) = 0$ such that the global minimizer of $\phi_Q(\mu)$ is attained at $\mu = 0$, provided the matrix has a full rank, which is also the least squares solution itself. Golub et. al. [19] point out that there is usually a large maximum in the close vicinity of $\mu = 0$ and the desired minimum of $\phi_Q(\mu)$ lies to the right of this maximum. Typically, a quasi-optimality criterion curve has multiple local minimums and makes it difficult to find the desired regularization parameter.

2.5.3 Generalized Cross-Validation (GCV)

Generalized cross-validation as a method for choosing the regularization parameter was originally proposed by Golub et. al [20]. The philosophy of cross-validation is based on the leave-one-out idea. Omitting the k^{th} observation y_k produces a corresponding leave-one-out solution vector x_μ^k. This solution can be used to predict the "missing" observation, y^k. A good regularization parameter will produce a solution where the misfit $\big((Hx_\mu^k)_k - y_K\big)$ is the low average over all possible y_k.

The GCV regularization parameter μ is determined as a global minimizer of,

$$\phi_{GCV}(\mu) := min \left\{ \frac{\|(\bar{H}\bar{H}^T + \mu I)^{-1} y\|}{trace\left(\bar{H}\bar{H}^T + \mu I\right)^{-1}} \right\} \tag{2.21}$$

The trace term in the denominator of $\phi_{GCV}(\mu)$ may not be feasible for evaluating large matrices $\bar{H}$. In such cases stochastic trace estimators can be introduced to approximate the trace [19]. Let U be a discrete random variable which takes the values +1 and −1 each with a probability of $\frac{1}{2}$, and let u be a vector of size m whose entries are independent samples from U. Then,

$$\tilde{t}(\mu) = u^T(\bar{H}\bar{H}^T + \mu I)^{-1}u \tag{2.22}$$

is an unbiased estimator of

$$t(\mu) = trace(\bar{H}\bar{H}^T + \mu I)^{-1}$$

Thus we only consider the minimization of the stochastic GCV function

$$\tilde{\phi}_{GCV}(\mu) := \frac{\sqrt{y^T(\bar{H}\bar{H}^T + \mu I)^{-2}y}}{u^T(\bar{H}\bar{H}^T + \mu I)^{-1}u} \tag{2.23}$$

The other method to calculate the GCV curve is to use SVD of $\bar{H}$. Let $\bar{H} = U\Sigma V^T$ denote the SVD of $\bar{H}$, where the u_i of U and v_i of V contain, respectively, the left and right singular vectors of $\bar{H}$, and $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is the diagonal matrix containing the singular values of $\bar{H}$, with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$. Chung et. at. [9] describes regularization parameter μ as the global minimizer of the GCV curve defined by,

$$\tilde{\phi}_{GCV}(\mu) = \frac{n\left(\sum_{i=1}^{n}\left(\frac{\mu u_i^T y}{\sigma_i^2 + \mu}\right)^2 + \sum_{i=n+1}^{m}(u_i^T y)^2\right)}{\left((m-n) + \sum_{i=1}^{n}\frac{\mu}{\sigma_i^2 + \mu}\right)} \tag{2.24}$$

Both the methods described to calculate GCV curves above are equivalent.

2.5.4 The L-curve Criterion

Lawson and Hanson [36] observed in the '60s that an efficient way to display how the value of the regularization parameter affects the solution x_μ of equation (2.15) and the error $(Hx_\mu - y)$, is to plot the curve defined by

$$\begin{aligned} \pounds &= (log\|Mx_\mu\|, log\|Hx_\mu - y\|) : \mu \geq 0 \\ & \quad or \\ \pounds &= \big(log\|\bar{x}_\mu\|, log\|\bar{H}\bar{x}_\mu - y\|\big) : \mu \geq 0 \end{aligned} \tag{2.25}$$

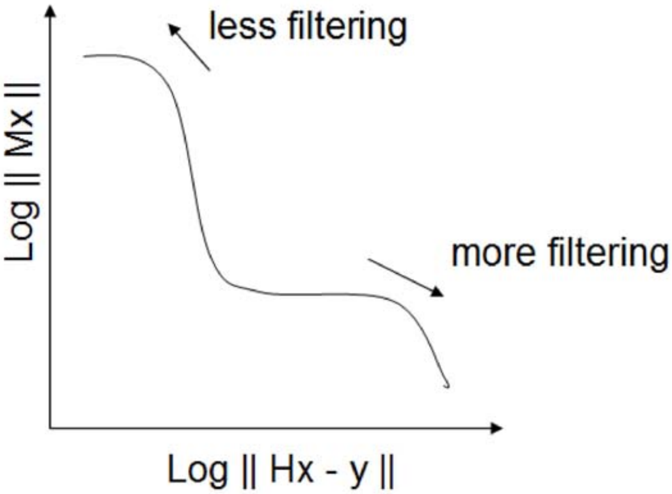


Figure 2.2: L-curve illustration(Hansen [23])

This curve is commonly referred to as the L-Curve, because for many linear discrete ill-posed problems with a right-hand side contaminated by error, the graph of the curve looks like the letter “L” e.g Figure (2.2). The L-curve clearly displays the compromise between minimization of these two quantities, which is the heart of any regularization method. The L-curve divides the first quadrant into two regions. It is impossible to construct any solution that corresponds to a point below the Tikhonov L-curve. Any regularized solution must lie on or above this curve. For a small μ the behavior of the L-curve is entirely dominated by the contributions from the observation errors e. This corresponds to the uppermost part of the curve in the figure (2.2) and the vertical part where $\|Mx_\mu\|$ is very sensitive to the changes in the regularization parameter. For large μ at the rightmost part of the L-curve, it is the residual norm $\|Hx_\mu - y\|$ that is most sensitive to the regularization parameter. We would like to reduce the residual norm without “blowing up” the solution norm. This defines the region in between the two extremes which is the L-shaped corner of the L-curve. This criterion can be described as maximizing the curvature of the L-curve. Reginska [41] and Hansen [25] proved that the log-log L-curve is always strictly concave for $\mu \leq \sigma_n$ (the smallest singular value of $\bar{H}$) and for $\mu \geq \sigma_1$ (the largest singular value of $\bar{H}$). Thus, the L-curve is always concave at its “ends” near the axis.

2.6 Projection methods

Using GCV or L-curve can be quite effective to solve the Tikhonov regularization problem. But, the GCV method requires us to employ the singular value decomposition of $\bar{H}$ or find a trace of the large matrix. The approach for the L-curve to determine a suitable value of the regularization parameter requires that one compute many points on the L-curve and evaluate the curvature of the L-curve at these points. These methods are prohibitively expensive for the problems where the size of the matrix H is large. In this study, the GRACE gravity solution problem typically has a parameter space of roughly 35000 parameters, for a degree order 180x180 gravity field. Employing the direct method described above to solve for the regularization parameter for the GRACE solution is not feasible. Thus, for big problems like GRACE, we usually project the problem onto a subspace of a smaller dimension using a procedure called Lanczos bidiagonalization (LBD). The Lanczos bidiagonalization procedure is a critical component for the regularization of the large estimation problems and is discussed in detail in the section (2.6.1). By using the projection method to find the bounds of the L-curve and the curvature in the case of the L-curve method, to find the regularization parameter, we can solve the large-scale ill-posed inverse problems more efficiently. Projection methods can also be used to find the optimal μ, using the bounds on the GCV curve or Quasi-optimality curve in the GCV or quasi-optimality method, respectively.

2.6.1 Lanczos bidiagonalization

As the scientific and engineering problems get larger, the direct methods that compute the decomposition of a matrix become too costly for certain applications. Hence there is a need for iterative methods to solve the linear systems of equations. Lanczos bidiagonalization is one such iterative algorithm that can be used in this study with the large least square estimation problem. This section will investigate the serial algorithm of the Lanczos bidiagonalization method.

The Lanczos bidiagonalization of $\bar{H}$ with starting vector $u_1 = \frac{y}{\|y\|}$ produces two sets of orthogonal Lanczos vectors $\{u_i\}$ and $\{v_i\}$ such that, after k steps and with

$$U_k = (u_1, u_2, \ldots, u_{k+1}), \quad V_k \;=\; (v_1, v_2, \ldots, v_k)$$

we obtain a $(k+1) \times k$ bidiagonal matrix $C_{k+1,k}$ satisfying

$$\bar{H}V_k = U_k C_{k+1,k} \tag{2.26}$$

The kth iteration of LBD computes the k^{th} column of V_k and $(k+1)^{st}$ column of U_k. Specifically, at the iteration k,

$$\begin{aligned} \bar{H}^T U_k &= V_k C_{k+1,k}^T + \alpha_{k+1} u_{k+1} e_{k+1}^T \\ \bar{H} V_k &= U_k C_{k+1,k} \end{aligned} \tag{2.27}$$

where, e_{k+1} denotes the last column of the identity matrix of dimension $(k+1)$ and α_{k+1} will be the $(k+1)^{st}$ diagonal entry of $C_{k+2,k+1}$.

Now, let the singular value decomposition of $C_{k+1,k}$ be denoted by

$$C_{k+1,k} = P_k \Sigma_k Q_k^T \tag{2.28}$$

The large singular values of $C_{k+1,k}$ given by σ_k, are approximation to the large singular values of $\bar{H}$. Moreover, approximations to the corresponding left and right singular vectors of $\bar{H}$ are given by the associated columns of the matrices $U_k P_k$ and $V_k Q_k$. As k increases, the accuracy of some of the approximations increase while other singular values in σ_k start approximating other singular values $\bar{H}$. In this way we can use Lanczos bidiagonalization to compute, in principle, as many singular values and vectors of $\bar{H}$ as desired. In the presence of round-off errors, the convergence properties of the Lanczos bidiagonalization have problems because the Lanczos vectors loose orthogonality. However, re-orthogonalization of these vectors solve most of these problems. The serial algorithm for k-steps of Lanczos bidiagonalization of $A = \bar{H}$ with the starting vector $y_0 = y$ is given in algorithm (2.1)

Algorithm 2.1 Serial Lanczos Bidialgonalization

```
Choose a starting vector $y_0 \in \Re^m$, and let
$\beta_1 = \|y_0\|_2$, $u_1 = y_0/\beta_1$ and $v_0 \equiv 0$
for $j = 1, 2, \cdots, k$ do
    $r_j = A^T u_j - \beta_j v_{j-1}$
    $\alpha_j = \|r_j\|_2$
    $v_j = r_j/\alpha_j$
    $p_j = A v_j - \alpha_j u_j$
    $\beta_{j+1} = \|p_j\|_2$
    $u_{j+1} = p_j/\beta_{j+1}$
end for
```

2.6.2 Parallel algorithm for Lanczos bidiagonalization

Appendix B discusses the parallelization concepts required to parallelize the Lanczos bidiagonalization algorithm. In this section, we will apply some of these concepts to parallelize the algorithm (2.1). This section follows the explanation and algorithm described in [34]. When we look at algorithm (2.1), we notice that the main computational work is in the two matrix vector products $v = A^T u - \beta v$ and $u = Av - \alpha u$. As a more general and representative problem, A is assumed to be a dense matrix. The matrix-vector multiplication operations, which are at the heart of the recursion in this algorithm are simple *BLAS-2* operations [34]. It should be noted that the entire A matrix is required for the computation of the Lanczos bidiagonal matrix, since we use both A and A^T in the same iteration step. Hence, a large problem like GRACE has a huge memory requirement and data distribution among the processors is necessary. Let us assume that there are P number of total processors and k number of iterations required. Further, assume that matrix $A \in \Re^{(m \times n)}$ can be partitioned into $M \times N$ rectangular blocks A_{ij} of size $m_b \times n_b$ such that $Mm_b = m$ and $Nn_b = n$ as:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MN} \end{bmatrix} \tag{2.29}$$

and the vectors $u \in \Re^m$ and $v \in \Re^n$ are partitioned accordingly:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{bmatrix} \quad , \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{2.30}$$

where $u_i \in \Re^{m_b}, i = 1, 2, \cdots, M$ and $v_i \in \Re^{n_b}, i = 1, 2, \cdots, N$

If the matrix in the problem is sparse, the partitioning can be chosen on the basis of the the non-zero pattern so that the load in the matrix-vector multiplication is well balanced. But in this case no consideration is made on the non-zero pattern of the matrix and the parallelization is performed assuming a dense matrix. If we try to parallelize the algorithm (2.1) as it is, we see that the block partition of A needs to be loaded into the processor cache twice to perform two matrix-vector operations ($A^T u$ and Av). And since the result of Av is used in the calculation of $A^T u$ and vice-versa, the processors would need to communicate the intermediate results to each other between the computations. But with simple reformulation of the recursion, it is possible to compute a segment of v_j and immediately afterwards use this intermediate result to accumulate a term in the sum for each segment in u_{j+1}. By doing this any given block A only needs to be loaded into the cache once per iteration.

We introduce the temporary vectors $xtemp \in \Re^n$ and $ytemp \in \Re^m$, where

$$xtemp = \begin{bmatrix} xtemp_1 \\ xtemp_2 \\ \vdots \\ xtemp_N \end{bmatrix} \quad , \quad ytemp = \begin{bmatrix} ytemp_1 \\ ytemp_2 \\ \vdots \\ ytemp_M \end{bmatrix}$$

and the modified version of the algorithm is give by:

We divide the matrix A among the processors by block rows such that $M = P$(total number of processors). N is such that the entire matrix A can fit in the available memory. The parallelization of the interleaved matrix-vector products for computing $x = A^T u - \beta v$ and $y = Ax$ in the algorithm (2.2) is given in algorithm (2.3)

Algorithm 2.2 Modified Lanczos Bidialgonalization

```
Choose a starting vector $y_0 \in \Re^m$, and let
$\beta_1 = \|y_0\|_2$, $u_1 = y_0/\beta_1$ and $v_0 \equiv 0$
for $j = 1, 2, \cdots, k$ do
    $xtemp = A^T u_j - \beta_j v_{j-1}$
    $ytemp = Ax$
    $\alpha_j = \|xtemp\|_2$
    $v_j = xtemp/\alpha_j$
    $ytemp = y/\alpha_j - \alpha_j u_j$
    $\beta_{j+1} = \|ytemp\|_2$
    $u_{j+1} = ytemp/\beta_{j+1}$
end for
```

Algorithm 2.3 Parallel Lanczos Bidialgonalization algorithm

```
for $p = 1$ to $P$ processors do
    $y_p = 0$
end for
for $q = 1$ to $N$ do
    for $p = 1$ to $P$ processors do
        $x_q^{local(p)} = A_{pq}^T u_p$
    end for
    accumulate $x_q = \sum_{j=1}^{P} x_q^{local(j)}$
    $x_q = x_q = \beta v_q$
    $\forall j \in 1 \cdots q : x_j = \sum_{k=1}^{M} A_{kj}^T u_k - \beta v_j$
    for $p = 1$ to $P$ processors do
        $y_p = y_p + A_{pq} x_q$
    end for
    $\forall j \in 1 \cdots M : y_j = \sum_{k=1}^{q} A_{jk} x_k$
end for
```

The algorithm 2.3 does not re-orthogonalize the left and right Lanczos vectors. Due to the orthogonality problems in the presence of round off errors, the Laczos bidiagonalization process should be re-orthogonalized. [35] discusses re-orthogonalization of the serial process which is applied to the modified Lanczos bidiagonalization algorithm (2.2). At each step of the Lanczos bidiagonalization process, v_{i+1} and u_{i+1} are orthogonalized against all the previous Lanczos vectors as listed in the algorithm (2.4). The algorithm (2.4), with algorithm (2.3) to compute the matrix vector products, gives a parallelized Lanczos bidiagonalization with re-orthogonalization.

Algorithm 2.4 Modified Lanczos Bidialgonalization with reorthogonalization

```
Choose a starting vector $y_0 \in \Re^m$, and let
$\beta_1 = \|y_0\|_2$, $u_1 = y_0/\beta_1$ and $v_0 \equiv 0$
for $j = 1, 2, \cdots, k$ do
    $xtemp = A^T u_j - \beta_j v_{j-1}$
    re-orthogonalize
    for $i = 1$ to $j - 1$ do
        $xtemp = xtemp - (v_i^T xtemp) v_i$
    end for
    $ytemp = Ax$
    $\alpha_j = \|xtemp\|_2$
    $v_j = xtemp/\alpha_j$
    $ytemp = y/\alpha_j - \alpha_j u_j$
    re-orthogonalize
    for $i = 1$ to $j$ do
        $ytemp = ytemp - (u_i^T ytemp) u_i$
    end for
    $\beta_{j+1} = \|ytemp\|_2$
    $u_{j+1} = ytemp/\beta_{j+1}$
end for
```

2.6.3 L-Ribbon and curvature-ribbon

This method uses Lanczos bidiagonalization to find the bounds in the L-curve and its curvature [6]. The solution to the Tikhonov regularization problem is given by the equation (2.15) as

$$\begin{aligned} \bar{x}_\mu &= (\bar{H}^T\bar{H} + \mu I)^{-1}\bar{H}^T y \\ & \quad or \\ x_\mu &= (H^T H + \mu M^T M)^{-1} H^T y \end{aligned}$$

where $\bar{H} = HM^{-1}$ and $\bar{x} = Mx$

If we introduce,

$$\begin{aligned} \eta_\mu &= \|\bar{x_\mu}\|^2 = \|Mx_\mu\|^2 \\ & \quad and \\ \rho_\mu &= \|\bar{H}\bar{x_\mu} - y\|^2 = \|Hx_\mu - y\|^2 \end{aligned}$$

It can be shown that,

$$\begin{aligned} \eta_\mu &= \|\bar{x_\mu}\|^2 = y^T\bar{H}(\bar{H}^T\bar{H} + \mu I)^{-2}\bar{H}^T y \\ \rho_\mu &= \|\bar{H}\bar{x_\mu} - y\|^2 = \mu^2 y^T(\bar{H}\bar{H}^T + \mu I)^{-2} y \end{aligned} \tag{2.31}$$

Let,

$$\begin{aligned} \hat{\eta}_\mu &:= \log \eta_\mu \\ \hat{\rho}_\mu &:= \log \rho_\mu \end{aligned} \tag{2.32}$$

then the point of the L-curve associated with the value of μ of the regularization parameter is given by $P_\mu = \frac{1}{2}(\hat{\eta}_\mu, \hat{\rho}_\mu)$. The curvature of the L-curve at P_μ is

$$\kappa_\mu = \frac{\hat{\rho}'_\mu \hat{\eta}''_\mu - \hat{\rho}''_\mu \hat{\eta}'_\mu}{\left((\hat{\rho}'_\mu)^2 + (\hat{\eta}'_\mu)^2\right)^{\frac{3}{2}}} \tag{2.33}$$

where $'$ denotes differentiation with respect to μ. It follows from the equations (2.32) and $\hat{\rho}'_\mu = -\mu \hat{\eta}'_\mu$ that

$$\kappa_\mu = 2\frac{\eta_\mu \rho_\mu}{\eta'_\mu} \frac{\mu \eta'_\mu \rho_\mu + \eta_\mu \rho_\mu + \mu^2 \eta_\mu \eta''_\mu}{\mu^2 \eta_\mu^2 + \rho_\mu^2} \tag{2.34}$$

It follows from equation (2.31)

$$\eta'_\mu = -2y^T \bar{H}(\bar{H}^T \bar{H} + \mu I)^{-3} \bar{H}^T y \tag{2.35}$$

The equations (2.31) and (2.35) are used to derive the bounds on the L-Curve and its curvature. This method utilizes k steps of the Lanczos Bidiagonalization of the matrix $\bar{H} = HM^{-1}$ with the initial vector y. This yields a $(k+1) \times k$ bidiagonal matrix $C_{k+1,k}$ whose leading $k \times k$ submatrix is defined as C_k. QR factorization of $C_{k+1,k}$ yields a $k \times k$ matrix $\hat{C}_k$. Let $\hat{C}_{k,k-1}$ denote the $k \times (k-1)$ sub-matrix of $\hat{C}_k$. The lower and upper bounds on η_μ, ρ_μ and η'_μ the derivative of η_μ, are derived by substituting the Lanczos decomposition equation (2.26) into the right hand sides of equations (2.31) and (2.35). The connection between the Lanczos decomposition (2.26) and Gauss quadrature applied to judiciously chosen matrix functionals can be used to show that the derived quantities below indeed are upper and lower bounds [5]. The details of the derivation of these bounds are given in appendix C.

$$\rho_\mu^- = \mu^2 \|b\|^2 e_1^T (C_k C_k^T + \mu I_k)^{-2} e_1 \tag{2.36}$$

$$\rho_\mu^+ = \mu^2 \|b\|^2 e_1^T (C_{k+1} C_{k+1}^T + \mu I_{k+1})^{-2} e_1 \tag{2.37}$$

$$\eta_\mu^- = \|\bar{H}^T y\|^2 e_1^T (\hat{C}_k \hat{C}_k^T + \mu I_k)^{-2} e_1 \tag{2.38}$$

$$\eta_\mu^+ = \|\bar{H}^T y\|^2 e_1^T (\hat{C}_{k,k-1} \hat{C}_{k,k-1}^T + \mu I_k)^{-2} e_1 \tag{2.39}$$

$$(\eta_\mu')^- = -2\|\bar{H}^T y\|^2 e_1^T (\hat{C}_{k,k-1} \hat{C}_{k,k-1}^T + \mu I_k)^{-3} e_1 \tag{2.40}$$

$$(\eta_\mu')^+ = -2\|\bar{H}^T y\|^2 e_1^T (\hat{C}_k \hat{C}_k^T + \mu I_k)^{-3} e_1 \tag{2.41}$$

where e_j is the j^{th} axis vector and I_k is the $k \times k$ identity matrix.

One should note that the $z_\mu := (C_k C_k^T + \mu I_k)^{-1} e_1$ needed to solve for the bound in equation (2.36) (and similarly in the other equations), can be solved as a least squares problem with a QR factorization. Using equations (2.36) - (2.41) and (2.34), we plot the bounds on the L-curve by plotting $\log \rho_\mu^\pm$ v/s $\log \eta_\mu^\pm$ and the bounds on its curvature with $\kappa_\mu^\pm$ v/s μ. The μ value corresponding to the maximum in the curvature plot is usually chosen as the regularization parameter μ for this L-ribbon method. Since we project the larger problem onto a much smaller subspace, it becomes computationally inexpensive to solve a problem of any size.

2.6.4 Bounds for other projection methods

The regularization parameters in GCV and quasi-optimality methods can also be solved by projecting the large regularization problem on a much smaller problem, using Lanczos bidiagonalization method. Using the bidiagonal projected problem we can find the bounds on the GCV and quasi-optimality curves in order to find the largest global minimizer μ. The derivation for the bounds are given in the appendix (C).

In the case of Quasi optimality condition, the bound on the equation (2.20) is given by,

$$\begin{aligned}\phi_Q^+(\mu) &:= \mu^2\|\bar{H}^T y\|^2 e_1^T(\hat{C}_{k,k-1}\hat{C}_{k,k-1}^T + \mu I)^{-4} e_1 \\ \phi_Q^-(\mu) &:= \mu^2\|\bar{H}^T y\|^2 e_1^T(\hat{C}_k\hat{C}_k^T + \mu I)^{-4} e_1 \end{aligned} \tag{2.42}$$

where e_j is the j^{th} axis vector and I_k is the $k \times k$ identity matrix and $\hat{C}$ is as defined in section (2.6.3).

Similarly, in the case of the projection of the GCV problem using Lanczos bidiagonalization, we get the bounds on the GCV curve as defined by the equation (2.23)as follows

$$\begin{aligned}\tilde{\phi}_{GCV}^+(\mu) &:= \frac{\sqrt{\|y\|^2 e_1^T(C_{k+1}C_{k+1}^T + \mu I)^{-2} e_1}}{u^T(C_kC_k^T + \mu I)^{-1}u} \\ \tilde{\phi}_{GCV}^-(\mu) &:= \frac{\sqrt{\|y\|^2 e_1^T(C_kC_k^T + \mu I)^{-2} e_1}}{u^T(C_{k+1}C_{k+1}^T + \mu I)^{-1}u} \end{aligned} \tag{2.43}$$

Like in section (2.6.3), the $z_\mu := (C_kC_k^T + \mu I_k)^{-1}e_1$ needed to solve for the bounds in the equation above, can be solved as a least squares problem with a QR factorization

2.7 Regularization of GRACE estimation problem

The equations used to solve the least squares problem with regularization in equation (2.15) uses the the matrix H, which is the matrix of the partial derivative of the observations with respect to the parameters to be estimated, mapped to a single epoch. This matrix is of the size $m \times n$, where m is the number of observations and n is the number of parameters to be estimated. In the case of the grace estimation problem for the monthly gravity field, the typical size of such an H matrix for a degree and order 120×120 problem would roughly be $2,000,000 \times 15,000$. Handling such a large matrix for computations for regularization would be unrealistic.

GRACE current release (RL04) employs orthogonal transformations as described in section (A.3) to convert the matrix H of size $m \times n$ to an upper triangular matrix of size $n \times n$. The corresponding vector y of size m gets transformed into a vector b of size n by the orthogonal transformations. The matrix R and the vector b is used to solve for the estimate $\hat{x}$, as derived in section (A.3). These upper triangular matrices of size $n \times n$ are saved during the processing of the GRACE data and will be used for regularization of the GRACE solution instead of forming the H matrix. Section (2.7.2) shows the equivalence of using the orthogonal transformation matrix R instead of the full partials matrix H in the regularization process.

2.7.1 Parameter leveling for GRACE

The method of *normal equations* is one technique commonly used to solve least square systems. An alternative approach that uses orthogonal transformation, also known as QR factorization, is used for the solution in GRACE processing and is discussed in appendix (A). This approach based on QR factorization is used to avoid some of the numerical problems encountered in the normal equations approach. In order to use this method effectively and efficiently, the parameters are treated in the local, common or global context. This structure is referred to as *parameter leveling*. For a typical GRACE solution, three data files are used for each arc. An arc is defined as a specific length of time, typically one day. One file contains the information arrays (measurement residuals and partials of the observation with respect to the parameters to be estimated) for SST data while the other two contain the information arrays for GPS data for each of two GRACE satellites. The parameter space is divided into three types:

- **Local :** These are the parameters that are valid only for one arc and for a particular data set.
- **Common :** These are the parameters that are valid only for one arc but are common across the datasets for that arc.
- **Global :** These are the parameters that are common across all the arcs and all the datasets. This parameter set contains the geo-potential parameters to be estimated.

In order to obtain this *parameter leveling*, a considerable amount of book-keeping in needed to ensure that the parameters are aligned properly as each new data file is processed. Gunter, 2000 [21] gives an excellent overview of how this QR factorization works while updating the R matrix with a certain amount of data at a time. If the amount of data and number of parameters were small, it would be possible to form the entire H matrix and perform a standard QR factorization on this single matrix. The reason it is done by combining a batch of data at a time has to do with the fact that the R matrix is upper triangular and roughly half of its elements are zero. Since there is no correlation between the local parameters of different data sets or between common parameters from different arcs, the H matrix formed is also very sparse with large number of zeros and would involve a large number of unnecessary operations with zeros. Also, for the number of parameters that a typical GRACE problem determines and the number of observations used are so large that the entire H matrix would not fit into the memory for any processing, even on the massively parallel machines.

In order to visualize this process, consider an example with 2 arcs which has 2 data sets for each arc (eg. GPS and KBR). The set of parameters that are unique to dataset 1 of arc 1 will be called L_{11} and that of dataset 2 of arc 1 will be called L_{12} and so on. These parameters are classified as local parameters. The set of parameters that are common to both data set of arc 1 are called C_1 (similarly C_2 for common parameter set of arc 2). The set of parameters that are common to all the datasets and all the arc are called global parameters designated by G. For the sake of illustration, if memory was not an issue and if we form the entire H matrix of size $m \times n$, it would look like the illustration in figure 2.3, where the white blocks suggest zeros. Using the orthogonal transformation the upper triangular matrix R of size $n \times n$ is

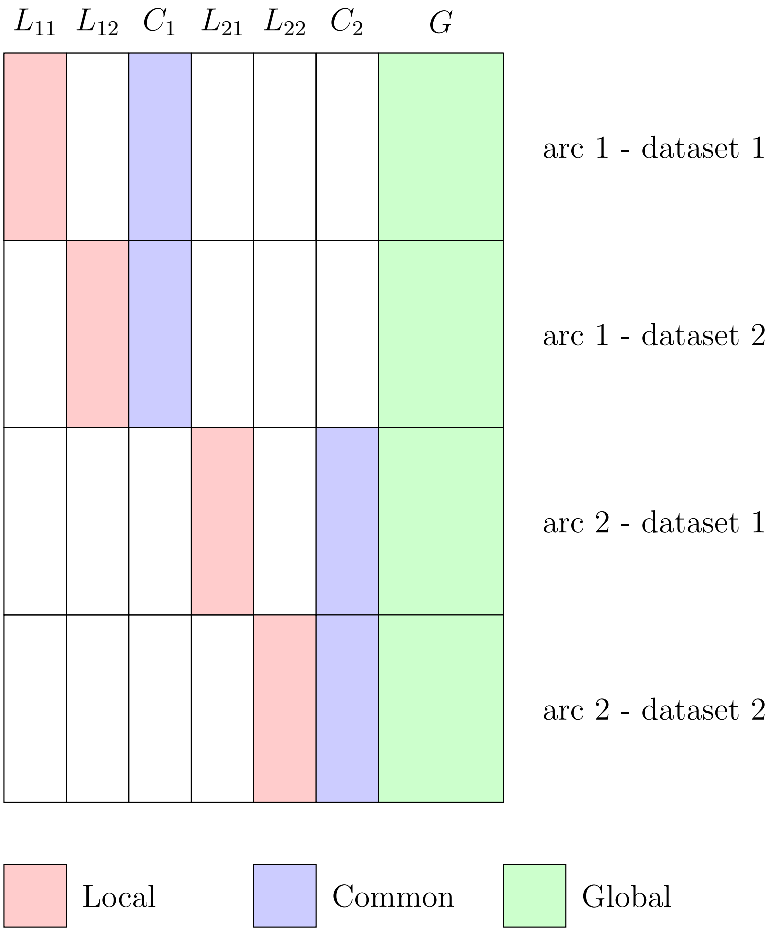


Figure 2.3: Parameter leveling for the full H matrix - An example with 2 arcs with 2 datasets each

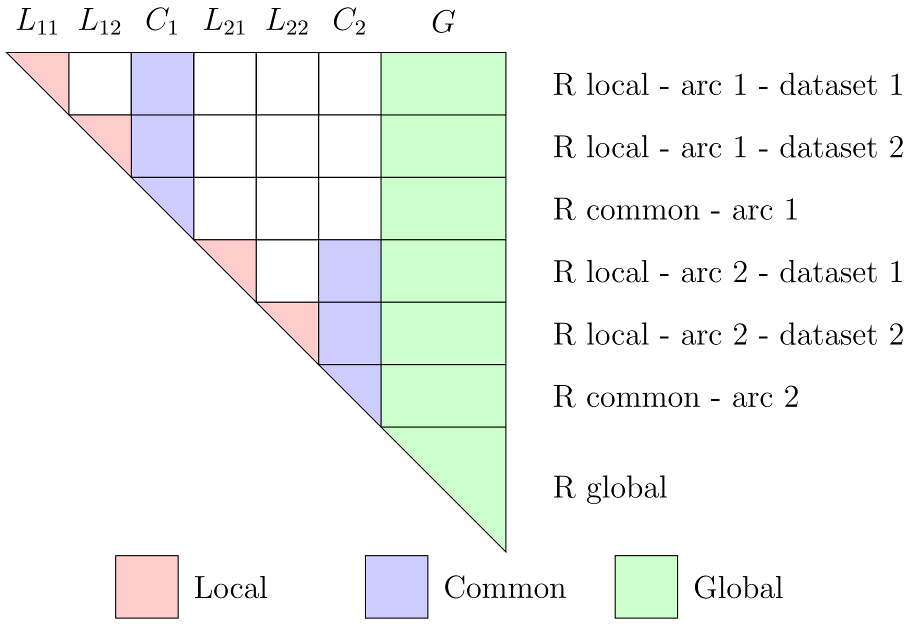


Figure 2.4: Parameter leveling for the full R matrix - An example with 2 arcs with 2 datasets each

computed which is illustrated by the figure 2.4. As shown in the appendix A, this R matrix is used to solve for the parameters using a backward substitution method. Thus, by only using the *R Global* part of the full R matrix, one can solve for all the global parameters including the geo-potential coefficients.

2.7.2 Orthogonal transformation and Tikhonov regularization

We can rewrite the Tikhonov regularization performance index as given in equation (2.13) as follows.

$$J(x) = min\{(Hx - y)^T(Hx - y) + \mu(x^T M^T Mx) \tag{2.44}$$

Following the derivation of the orthogonal transformation solution of the least square problem given in section (A.3), orthogonal matrix Q is selected such that,

$$\begin{aligned} QH &= \begin{bmatrix} R \\ O \end{bmatrix} \\ \text{and define } \; Qy &= \begin{bmatrix} \mathbf{b} \\ e \end{bmatrix} \end{aligned} \tag{2.45}$$

where, R is a $n \times n$ upper-triangular matrix of rank n

O is a $(m - n) \times n$ null matrix

$\mathbf{b}$ is a $n \times 1$ column vector

e is a $(m - n) \times 1$ column vector

Thus equation (2.44) becomes,

$$\begin{aligned} J(x) &= (Hx - y)^T Q^T Q(Hx - y) + \mu(x^T M^T Mx) \\ &= \|Q(Hx - y)\|^2 + \mu\|Mx\|^2 \end{aligned} \tag{2.46}$$

Since $m > n$ and H is of rank n, we can use equation(2.45) to rewrite the equation (2.46) as

$$J(x) = \left\| \begin{bmatrix} R \\ O \end{bmatrix} x - \begin{bmatrix} \mathbf{b} \\ e \end{bmatrix} \right\|^2 + \mu \|Mx\|^2 \tag{2.47}$$

which leads to

$$J(x) = \|Rx - \mathbf{b}\|^2 + \|e\|^2 + \mu\|Mx\|^2 \tag{2.48}$$

Since only the first and the last terms in equation (2.48) are functions of x, the value of x that minimizes $J(x)$ is also the one that minimizes

$$\tilde{J}(x) = \|Rx - \mathbf{b}\|^2 + \mu\|Mx\|^2 \tag{2.49}$$

The equation (2.49) is similar to the equation we started with, equation (2.44). It can thus be said that regularizing a least squares problem starting with H and y will yield the same result as regularizing the solution starting with its orthogonal trasformations, i.e. R and b, as shown above.

Equation (2.44) leads to the unique solution given by,

$$x_\mu = (H^T H + \mu M^T M)^{-1} H^T y \tag{2.50}$$

and equation (2.49) leads to the unique solution,

$$x_\mu = (R^T R + \mu M^T M)^{-1} R^T b \tag{2.51}$$

If we apply the orthogonal transformation $QH = \begin{bmatrix} R \\ O \end{bmatrix}$ and $Qy = \begin{bmatrix} \mathbf{b} \\ e \end{bmatrix}$ to equation (2.50), we can show that equation (2.50) is mathematically equivalent to equation (2.51). We have thus reduced the size of the regularization problem significantly by applying orthogonal transformation, but is still to the order of 15000×15000 for a degree and order 120 estimation problem. By applying the M^{-1} to R and b as discussed in section (2.4) we can write the equation (2.51) in the standard Tikhonov form as,

$$\bar{x}_\mu = (\bar{R}^T \bar{R} + \mu I)^{-1} \bar{R}^T b \tag{2.52}$$

where, $\bar{R} = RM^{-1}$ and $\bar{x} = Mx$. If the weights W are applied as in the equation (A.24), we would get the same result as equation (2.51) because the H matrix in A.24 is multiplied by the corresponding weights in W before the R matrix is accumulated by QR factorization. Thus the upper triangular R matrix already factors in the weights applied to the observations during the accumulation process.

2.7.3 Parameter choice methods for GRACE

The application of SVD methods to solve for GRACE gravity solutions has already been studied in detail by Hinga [26]. Also, the SVD methods for a large problem like GRACE is computationally expensive. Thus, an alternative method for regularization that is computationally inexpensive was desired. Tikhonov regularization provides an extra dimension that can be controlled during the solution process, which is the regularization matrix M. The design of the regularization matrix used in this study is discussed in chapter (3). Thus, Tikhonov regularization in combination with projection methods to find the regularization parameters provided ideal choices for regularizing the GRACE solutions. In addition, section 2.7.1 showed that after accumulation of the R

matrix, one can use only the *R Global* to compute the geo-potential parameters and the value of the estimates of the other local and common parameters does not affect the geo-potential parameters. Thus, while implementing the QR version of the Tikhonov regularization to GRACE we use only the *R Global* part of the full R to regularize the geo-potential parameters. Thus the optimal weighting has already been factored in the calculations during the formation of the R matrix. Figure (2.5) illustrates the entire flow of the GRACE regularization process.

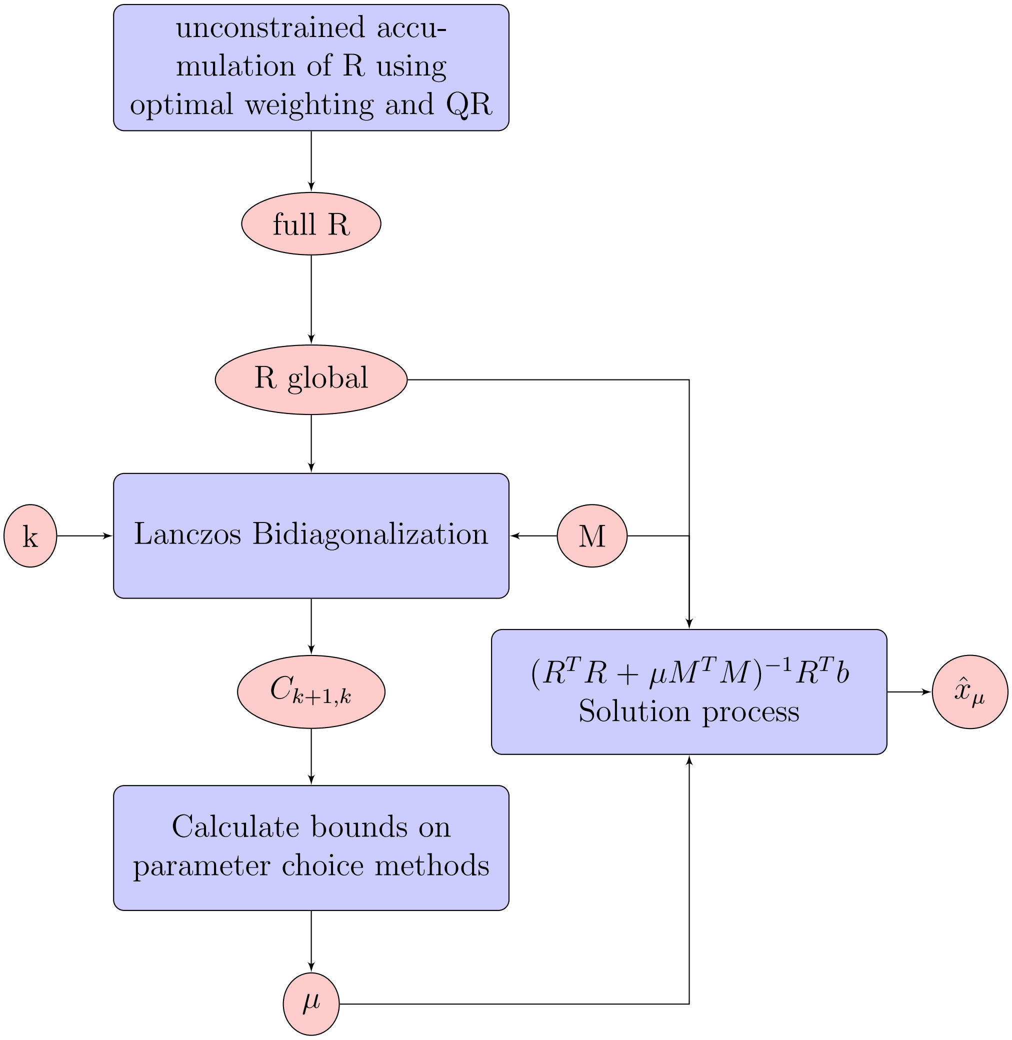


Figure 2.5: Flow chart of the regularization process for GRACE

Chapter 3

Design of the regularization matrix

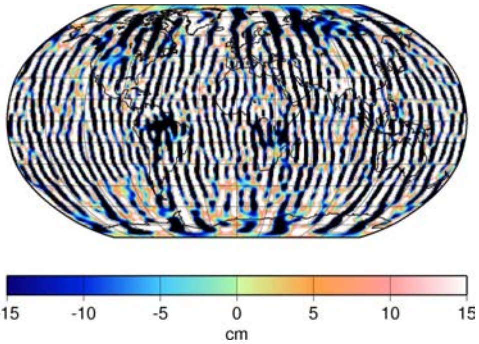
3.1 Errors in the GRACE RL04 solutions

The low degree and order harmonic co-efficients describe most of the time variable gravity signal of the Earth in the spherical harmonic equation (A.1). These co-efficients were not estimated very well before the initial GRACE results were published [50]. The high degree and order harmonics have a very slight effect on the satellite's orbit. The effects of these parameters attenuate rapidly with increasing altitude and the satellite tracking data observes only very small effects. These mid and high degree and order harmonics are vulnerable to unwanted signals from noise and other undesirable sources. Thus, due to the nature of the gravity inversion process, systematic errors enter the time variable (monthly) gravity solution during the estimation process. These systematic errors affect mostly the mid and high degree and order coefficients and manifest themselves as "stripes" in the global map of mass flux in the RL04 GRACE estimates.

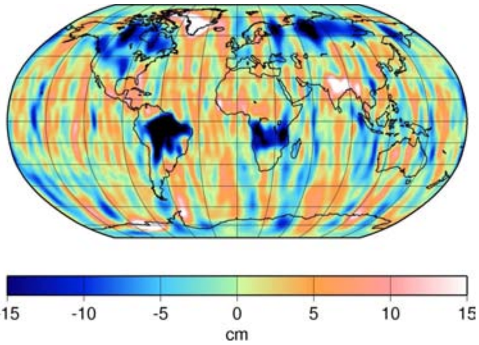
The least squares estimates ($\hat{x}$) from GRACE are the updates to the background models, which include the mean gravity field, atmospheric and ocean de-aliasing model, ocean tides, ocean pole tide and solid earth tides.

The corrections to the nominal reference model, ($\hat{x}$), capture the errors in the background models, the errors in the mean gravity field and the gravity change due to ice melt, hydrological and oceanic processes and glacial-isostatic adjustment (GIA). Monthly variability is defined as the deviation in the gravity field estimates with respect to a long term mean. Figure (3.1a) is a map of the monthly variability of equivalent water heights that demonstrates how these errors mask the signal. There are post-processing methods used in the literature to handle these errors. The most popular methods are Gaussian smoothing [57] and “de-striping” [46], applied before using the gravity solution. Figures (3.1b) and (3.1c) show the maps of mass flux in centimeter (cm) of equivalent water heights after applying a Gaussian smoothing of 350 km and 500 km respectively. The consequence of this is that errors are vastly reduced and the time variable signal is visible. The error stripes that are visible at 350 km smoothing are reduced even more with 500km smoothing. But this higher smoothing results in loss of signal and leakage between the basins [60].

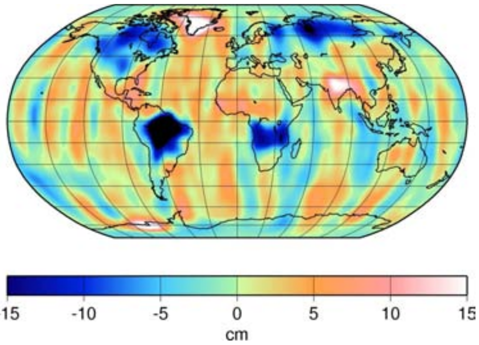
The errors in the GRACE solutions are correlated in the spectral domain. Examination of the Stokes coefficients for a particular order reveals the correlation [46]. The first two columns in the Figures (3.2) and (3.3) are for the month of April 2005 and the last two columns are for the month of May 2008. In figure (3.2), within the two columns for each month, the first column shows the Stokes coefficients, C_{lm}, as a function of degree (l) for the orders $m = 0$ through 3. In the second column for each month, the same Stokes coefficients are plotted separately as functions of even and odd degrees. Visual inspection



(a) May 2008 - no Gaussian smoothing



(b) May 2008 - 350 km Gaussian smoothing



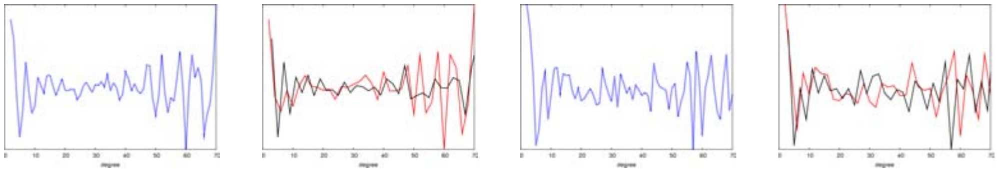
(c) May 2008 - 500 km Gaussian smoothing

Figure 3.1: Mass flux in the units of cm of equivalent water height using geopotential coefficients upto degree and order 60. Gaussian smoothing is applied as indicated

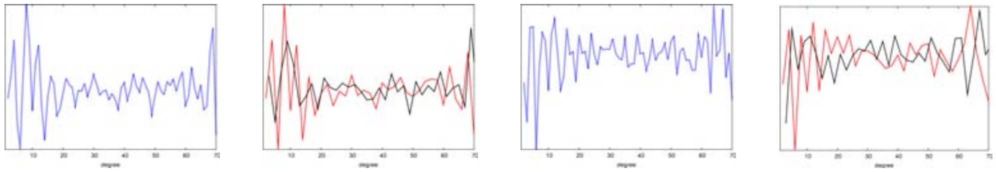
of these plots does not reveal any obvious correlations between the stokes coefficient as a function of the degree. Figure (3.3) is similar to the figure (3.2) for higher orders, $m = 8, 10, 14$ and 18. It can be seen that the Stokes coefficients of even or odd parity vary fairly smoothly. Even and odd coefficients do not appear correlated with one another [46]. This behavior begins approximately at order $m = 8$, and is present in all the higher orders. For a particular degree, there is no obvious correlation as a function of order. These figures suggest that the correlated errors enter the solution roughly around order 8. It can thus be said that, the coefficients below approximately degree 10 are virtually free of these correlated errors. Figure (3.3) also suggests that some months (eg May 2008) have correlated errors starting at a higher degree as compared to the some other months (eg April 2005).

3.2 Criteria for designing M

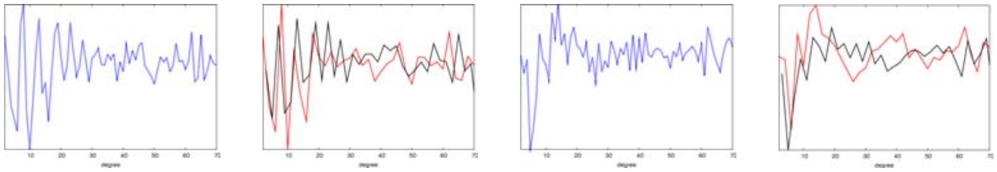
We have discussed the regularization process in the chapter (2) and discussed the options available to us for finding the regularization parameter. But the regularization process as given by the Tikhonov regularization equations (2.15) and (2.14), begins with the choice of a suitable regularization matrix M. The design of the regularization matrix is the key to supplementing the least squares problem with additional *pseudo-information* that makes the problem well conditioned. The regularization matrix also influences the choice of the regularization parameter μ when using the various methods outlined in chapter 2. We design the regularization matrix such that the resulting regularized



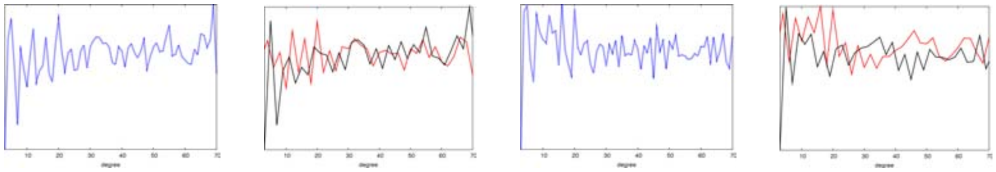
(a) Apr'05 - m=0 (b) Apr'05 - m=0 (c) May'08 - m=0 (d) May'08 - m=0



(e) Apr'05 - m=1 (f) Apr'05 - m=1 (g) May'08 - m=1 (h) May'08 - m=1

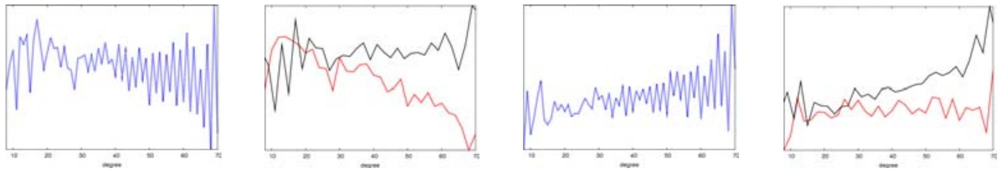


(i) Apr'05 - m=2 (j) Apr'05 - m=2 (k) May'08 - m=2 (l) May'08 - m=2

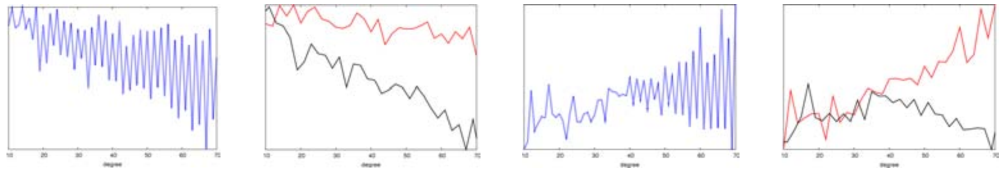


(m) Apr'05 - m=3 (n) Apr'05 - m=3 (o) May'08 - m=3 (p) May'08 - m=3

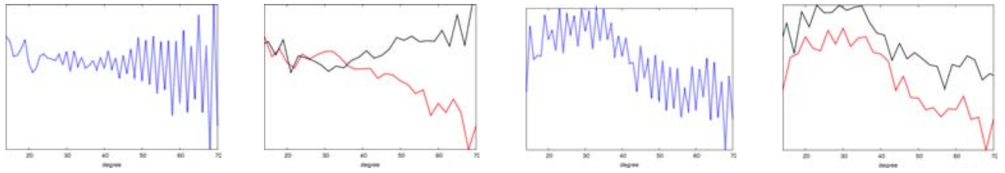
Figure 3.2: Stokes coefficient (C_{lm}), converted to mass, plotted as a function of degree for orders $m = 0, 1, 2$ and 3. April 2005 and May 2008



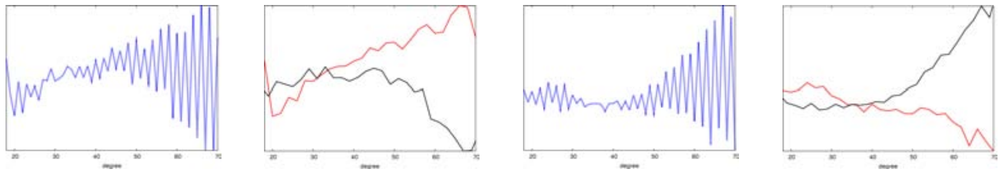
(a) Apr'05 - m=8 (b) Apr'05 - m=8 (c) May'08 - m=8 (d) May'08 - m=8



(e) Apr'05 - m=10 (f) Apr'05 - m=10 (g) May'08 - m=10 (h) May'08 - m=10



(i) Apr'05 - m=14 (j) Apr'05 - m=14 (k) May'08 - m=14 (l) May'08 - m=14



(m) Apr'05 - m=18 (n) Apr'05 - m=18 (o) May'08 - m=18 (p) May'08 - m=18

Figure 3.3: Stokes coefficient (C_{lm}), converted to mass, plotted as a function of degree for orders $m = 8, 10, 14, 18$. April 2005 and May 2008

solutions meet the following conditions.

- The regularized solutions fit the observations as well as an unconstrained solutions
- There is no attenuation of the signal in the regularized solutions when compared with the unconstrained solutions
- The regularized solutions show markedly reduced stripes (errors)
- The regularized solutions are of uniformly good quality over time

These conditions are subjective and, we assess success by observing the maps of geoid variability and post-fit data residuals. The post-fit residuals are obtained after fitting the solution to the observations by computing $(y - H\hat{x})$ for every observation $y(t)$. These post-fit residuals are a part of the observations that the parameterization fails to capture and gets discarded. Only the values of $\hat{x}$, which includes the geopotential parameters, are carried further for scientific analysis.

The post-fit residuals are passed through a differentiating low-pass CRN filter [61] that was designed specifically for GRACE data. CRN filter is a class of digital filter characterized by an $\mathbf{N}^{th}$ order self-**C**onvolution of **R**ectangular time-domain window function. This filter simultaneously differentiates and low-pass filters the range-rate into post-fit residual range accelerations. The benefit of this conversion is that the residuals are more easily visualized and assessed locally.

A map of these post-fit residuals show how well the solutions fit the data and highlight any attenuation of the signal. The time variable gravity signals in the regions like the Amazon, Africa, India, Greenland, Antarctica etc. have high magnitudes. If a map of these residuals show any geo-spatial correlations, specially in the regions with high signals, then that implies that the entire time-variable geopotential signal was not captured by the solution and that there may have been signal attenuation.

The maps of gravity field variability are used to determine the reduction in the errors in the solution. The reduced systematic errors in the solutions show up as reduced stripes in the variability maps. This exercise of designing the regularization matrix obtains a balance between the amount of noise acceptable in the solution without the reduction in the signal.

Since the error characteristics of the observations are different for different months, the regularization of every monthly solution can be considered as solving a new regularization problem. In order to maintain consistency throughout the mission for convenience, we design a single regularization matrix M that will satisfy the above conditions for all the months (or at least most of them).

The vector x in the Tikhonov minimization equation (2.14), is made up of all the parameters to be estimated including the deviation of the Stokes coefficients ($C_{l,m}$ and $S_{l,m}$) from the nominal gravity field, in addition to the other parameters like the initial conditions and other model parameters. Thus, a diagonal M in the Tikhonov minimization equation (2.13) provides constraints on the respective parameters in x which adds *pseudo information* to the inverse problem making the Tikhonov equation well-conditioned. The larger the entry in M, the more constrained the corresponding parameter will be. Since, the M needs to be invertible (chapter 2), the entires in M corresponding to the non gravity parameters can be set to a "small" but a non-zero entry, (e.g. 10^{-5}) and as a rule of thumb it should be a few orders of magnitude smaller than the smallest entry for the geo-parameters. This ensures that the non-gravity parameters are not constrained and are free to adjust to the data. If these parameters not allowed to adjust freely, they will introduce more errors in the gravity solution. In order to find the appropriate values for the entries in the regularization matrix (M) corresponding to the stokes coefficients, we can start with the most commonly used regularization matrix for gravity problems and then evolve at an acceptable regularization matrix by fixing the short comings of this matrix for the case of time variable gravity solution. The most commonly used regularization matrix for the gravity problems is the one which uses the "Kaula's rule" [10].

3.3 Definition of M using Kaula's rule

The most common regularization matrix used in the regularization of the gravity field estimation problem is the one populated using the Kaula's rule of thumb [33] [12]. Using the quantities defined in the spherical harmonic expansion given by the equation (A.1), we define a quantity, σ_l^2, called the "degree variances", by

$$\sigma_l^2 = \sum_{m=1}^{l}(c_{l,m}^2 + s_{l,m}^2) \tag{3.1}$$

where $c_{l,m}$ and $s_{l,m}$ are the uncertainties in the spherical harmonic co-efficients $C_{l,m}$ and $S_{l,m}$ respectively. From the studies of the covariance of surface gravity anomalies, William Kaula in 1966, approximated the magnitude of the square root of the degree variance σ_l^2 by the formula:

$$\sigma_l = \frac{10^{-5}\sqrt{2l+1}}{l^2} \tag{3.2}$$

This is known as the "Kaula's rule" and it provides an upper bound on the uncertainties in the spherical harmonic coefficients at high degrees. Using the equations (3.1) and (3.2), we can deduce that the upper bound on ($c_{l,m}$ and $s_{l,m}$) is $\frac{10^{-5}}{l^2}$. We can use the inverse of these bounds to populate a regularization matrix M. Since $\sqrt{\mu}$ scales the matrix M in the Tikhonov equation (2.13), we can ignore the constant 10^5 and let μ absorb it when we find its optimal value, using the methods outlined in chapter 2. We can now define the diagonal regularization matrix M_{kaula} as

$$\begin{aligned} M_{kaula}(i,j) &= 0 && \text{if } i \neq j \\ M_{kaula}(i,i) &= 10^{-3} && \text{non-geo paramters} \\ M_{kaula}(i,i) &= l^2 && \text{for } C_{l,m}/S_{l,m} \end{aligned} \tag{3.3}$$

Using this as a definition of the regularization matrix $M = M_{kaula}$, we find the values optimum values of μ using the L-ribbon method for all the months, as described in section (2.6.3). (More details about the parameter choices are discussed in the section (3.5)). Figure (3.4) shows the values of μ obtained using the L-ribbon method for the entire mission starting from Apr 2002 through July 2008, using the regularization matrix M_{Kaula}. Using these values for μ, the regularized solutions are estimated for the entire mission. The solution are tested to see if the requirements set in the section (3.2) are met with this choice of the regularization matrix.

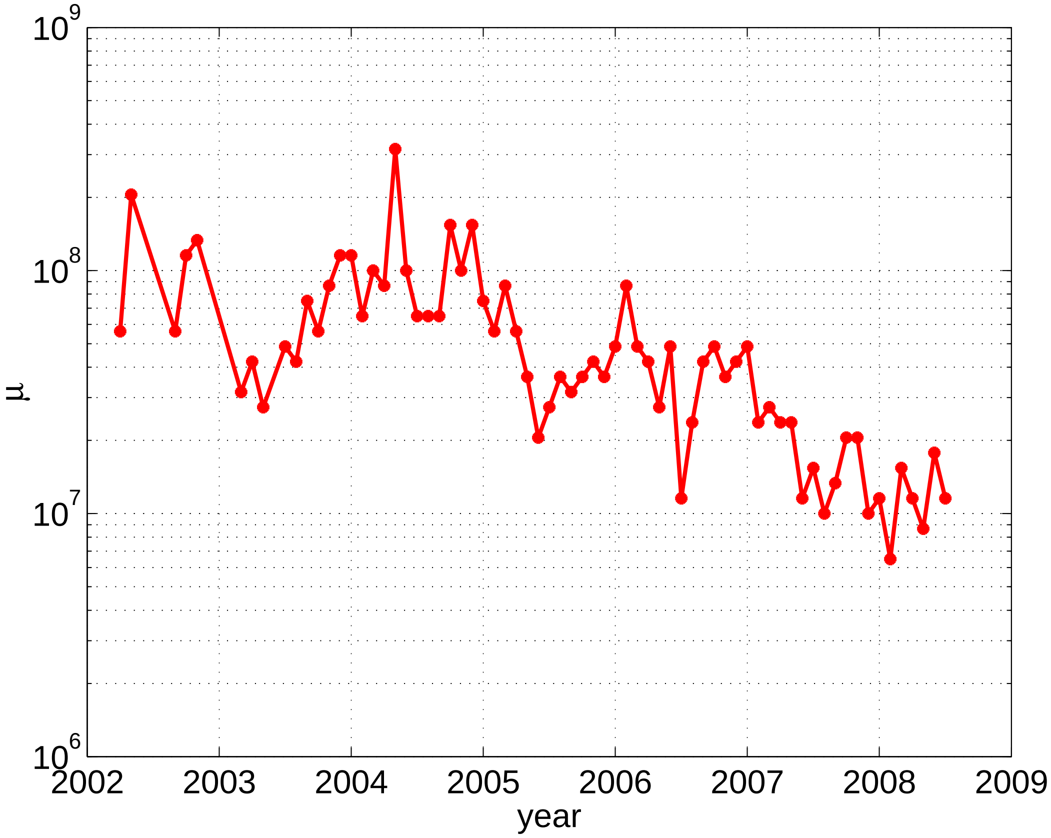
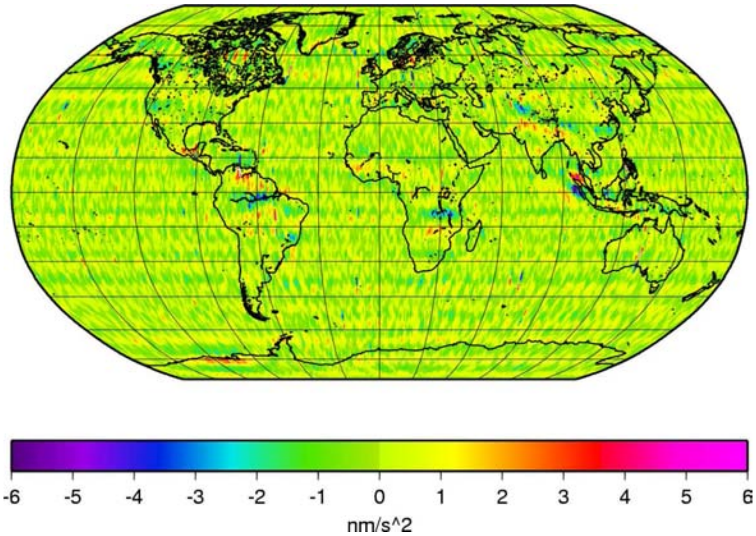
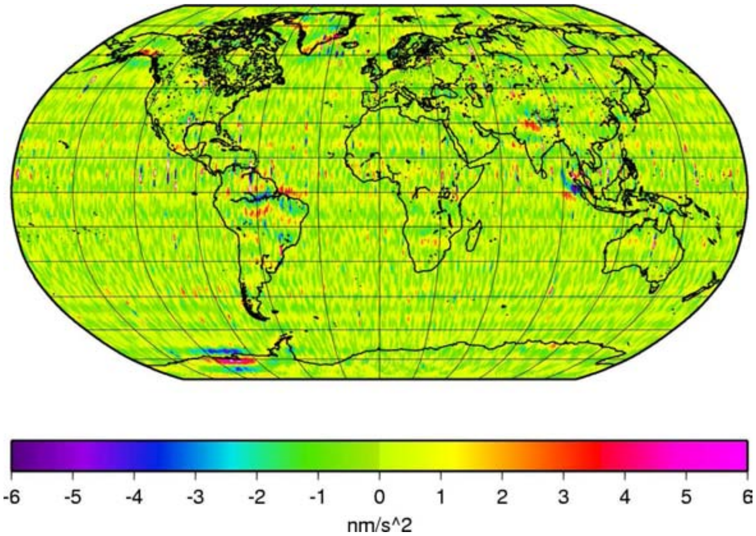


Figure 3.4: Plot of μ values for regularization using M_{kaula}

We compute the post-fit residuals as described in the section 3.2, con-

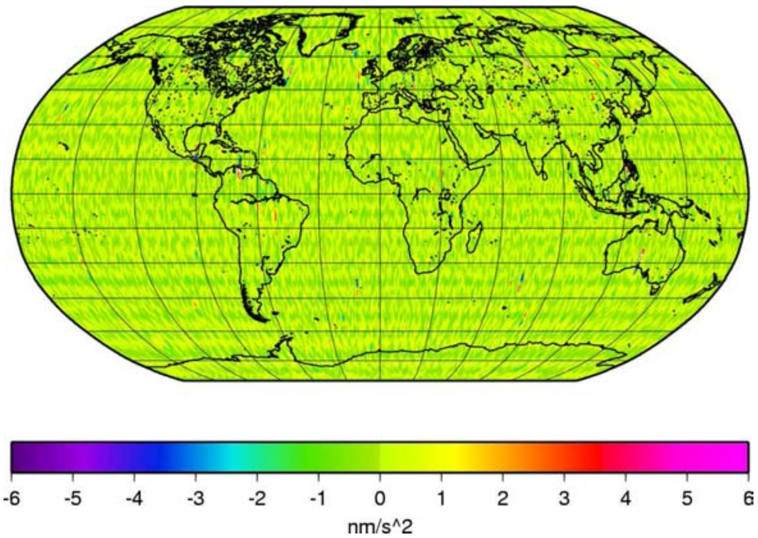


(a) Apr 2005

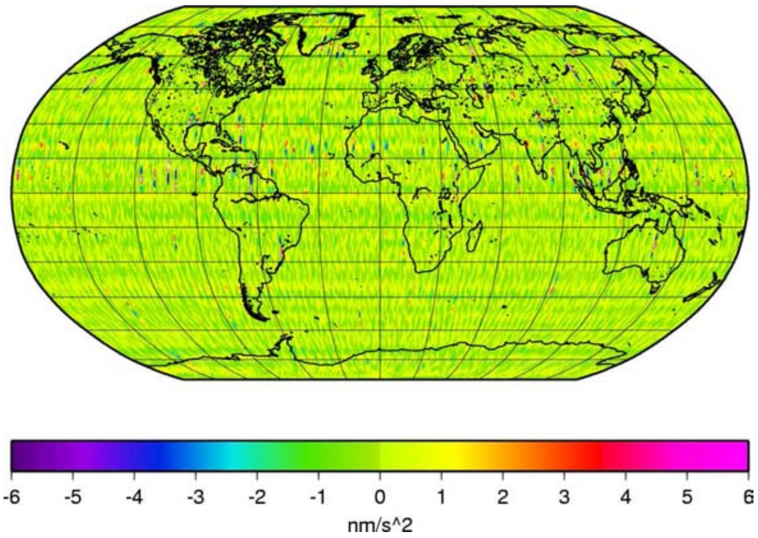


(b) May 2008

Figure 3.5: Residuals after fitting the Kaula based regularized solutions to the data in the units of range acceleration (nm/s^2)



(a) Apr 2005



(b) May 2008

Figure 3.6: Residual after fitting the RL04 solutions to the data in the units of range acceleration (nm/s^2)

vert them into the units of range-accelerations and plot them as a global map. Figures (3.5a) and (3.5b) are the maps of these residuals in the units of range-acceleration ($nanometers/s^2$) for the regularized solution of the months of Apr 2005 and May 2008 respectively using the Kaula regularization matrix (M_{kaula}). We can see a clear geo-spatial correlation in these residuals, specially in the regions of Antarctica, Greenland, Himalayas, Alaska and Amazon. This would suggest that all of the signal has not been captured by the regularized solutions using M_{kaula}. No such geo-spatial signal is visible at the same scale if we compare these residual maps to the figure (3.6a) and (3.6b), which are the residual maps for the unconstrained RL04 solutions. This is an evidence of a tight constraint on the parameters being estimated and thus not being able to capture signal in the mid degrees and high degrees, when using M_{Kaula} for regularization.

In the equation (2.15), $(\mu M^T M)^{-1}$ can be considered as the a priori covariance in equation (A.22) with the *apriori* solution $\bar{x} = 0$. This is not a statistical co-variance as used in the *apriori* case, but *pseudo-information* added to the performance index in equation (2.13) to make the problem well-conditioned. Thus, for the same M, a larger μ constrains the solution more to the nominal as compared to a smaller μ. We compare the value of $(\sqrt{\mu M^T M})_{i,i}$ for the two monthly solutions, Apr 2005 and May 2008, obtained using the Kaula regularization matrix M_{kaula}. Figure (3.7) is a plot of the $(\sqrt{\mu M^T_{Kaula} M_{Kaula}})_{i,i}$ v/s degree n, for order $m = 0$, $\forall$ degrees n, and i corresponding to geopotential coefficients only. In this figure, the greater the

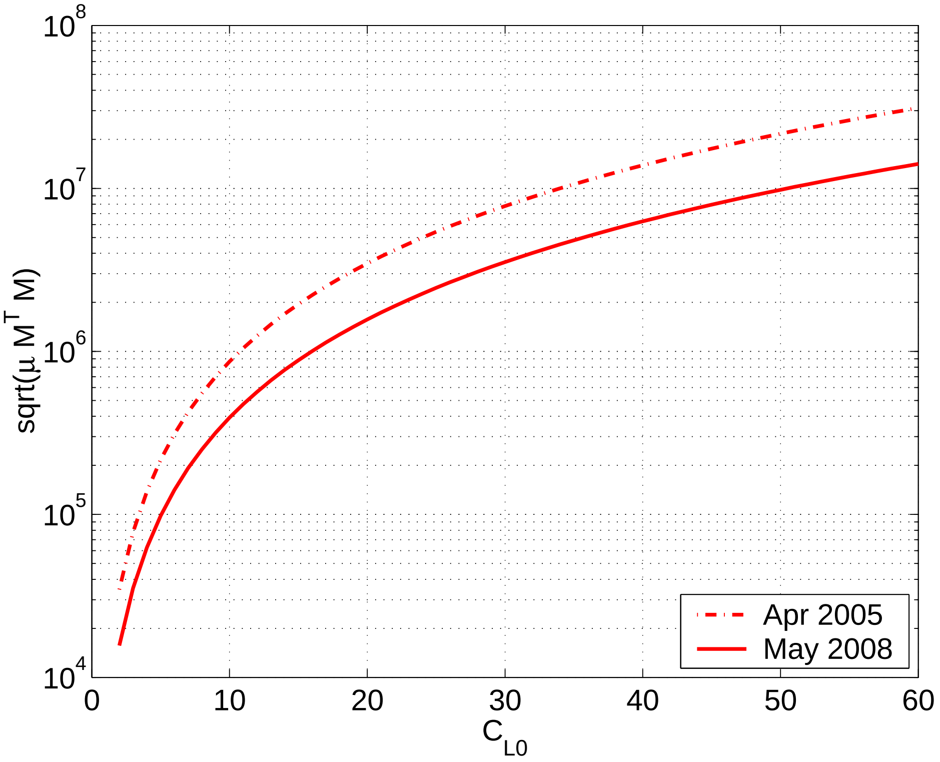


Figure 3.7: Plot of $\sqrt{\mu M^T M}$ values for C_{l0} for the solution of May 2004 and May 2008 using the regularization matrix M_{Kaula}

value of $(\sqrt{\mu M^T M})_{i,i}$, the more constrained the parameter is. By the definition of the regularization problem (equation 2.44), the value of μ obtained is influenced by the solution semi-norm $M\hat{x}$ as well as the residual $(y-H\hat{x})$. The systematic errors dominate different parts of the spectrum in different months and the value of $M(i,i)$ at these dominating spectra influence the value of the μ obtained as discussed in the section (3.1). The values in M_{kaula} are increasing with increasing degree. Thus, for two example months, Apr 2005 and May 2008, the value of μ obtained using the M_{kaula} regularization matrix, are far apart because of the different value of $M(i,i)$ at the spectrum with dominating errors for the two months. The values of μ obtained for these two months would be much closer to each other if the value of $(M(i,i))$ would be same over the different spectrum with dominating errors. As a consequence of this, the curve for Apr 2005 is higher than that for May 2008 in the figure (3.7). This results in the lower degrees for the month of May 2004 being much more constrained to the nominal gravity field than those for May 2008. Most of the variability signal is in the lower degrees and hence they should be the least constrained during regularization of the monthly gravity fields. Also, since the regularization matrix M_{kaula} is increasing with degree, the mid and high degree coefficients are constrained more than they should be. Because of these two reasons the gravity signal is being left in the residuals (figure (3.5)). The same analysis and comparison can be made with different orders m but it is sufficient to compare only one order, $(m = 0)$ as done in the figure (3.7)

3.4 Regularization matrix used for this study

The main drawback of the regularization matrix, M_{kaula} based on the Kaula's rule for time variable gravity, is that the matrix is increasing in the spectrum with errors which results in the high variations in the value of μ from one month to the other. Thus, it has a potential to constrain the solution at the lower degrees in some months and it always constrains the mid and high degrees. By fixing these problems we can generate a suitable regularization matrix that can be used for the regularization of the variability signal. Since the systematic errors enter the solutions at about degree 10, we choose to fix the value of $M(i,i)$ for all geo-parameters that correspond to the degrees greater than 10 to that of degree 10. This will ensure that the value of μ will vary less than what it did in the case of M_{Kaula} and it will also make sure that the solutions are not tightly constrained at the mid and high degrees. At the same time the value of $M(i,i)$ corresponding to the low degree coefficients should as low as possible to ensure that they are completely free to adjust to the data and capture all the time variable signal. Since the constraint achieved on the lower degrees is sensitive to the high variations in μ, which occurs due to the influence of the errors in the mid and high degrees, we need a much more steeper rise through the low degrees than that obtained by the matrix derived from Kaula's rule. Hence, the curve in the figure (3.7) corresponding to new regularization matrix should rise steeply though the low degrees and slow down to a constant value at degree 10. The steeper rise through the low degrees will make the constraint on the lower degrees less sensitive to the variation in μ from month to month. As discussed in the section (3.1), the lower order coefficients have no correlated errors. Hence, we introduce an order dependence in M, such that the lower orders are less constrained than the higher orders. Since the stripes characteristically have a sectoral structure, we constrain the sectoral somewhat more than the other coefficients.

The definition of the regularization matrix used in this study is as follows. Corresponding to every i^{th} element of the state vector $\hat{x}$, the $M(i,i)$ element of the diagonal regularization matrix M is populated with a specified value. The $M(i,i) := M_i$ diagonal element in M corresponding to the Stokes coefficients are defined as a function of the degree n and order m. Thus,

$$M_i = \Gamma(n,m) = \Psi(n)\Omega(m) \tag{3.4}$$

where,

$$\Psi(n) = \begin{cases} ((n+1)(n+2)(2n+1)/(1+k_n))^2 & \text{if } n \leq 4 \\ C_1((n+1)(n+2))^2 & \text{if } 4 < n \leq 7 \\ C_2(n+1)(n+2)(2n+1)/(1+k_l) & \text{if } 7 < n \leq 10 \\ \Psi(10) & \text{if } n > 10 \end{cases}$$

where k_n is the load love number of the degree n and C_1 and C_2 are constants so as to maintain continuity of the $\Psi(n)$ curve. The load love number is used in order to reduce the errors in the domain of the water-layer maps.

and,

$$\Omega(m) = \begin{cases} (m+1)^{0.1} & \text{if } m = 0 \\ (m+1)^{0.333} & \text{if } m = n \\ (m+1)^{0.2} & \text{if } m \neq n \end{cases}$$

M_i corresponding to the non-gravity parameters are set to 10^{-1} which is about 4 orders of magnitude smaller than the smallest value in M_i corresponding to the stokes coefficients. The regularization matrix obtained using this definition will be referred to as M_D in this study. Figure (3.8) plots the regularization matrix M_D in order to visualize the constraints achieved on the stokes coefficients while regularizing. The x-axis corresponds to degrees, the y-axis corresponds to the orders, with positive and negative y-axis plotting the values in M_D corresponding to C_{lm} and S_{lm} respectively. The z-axis and the color bar represent the magnitude of the M_i. The figure shows degree and order dependence in the regularization matrx M_D.

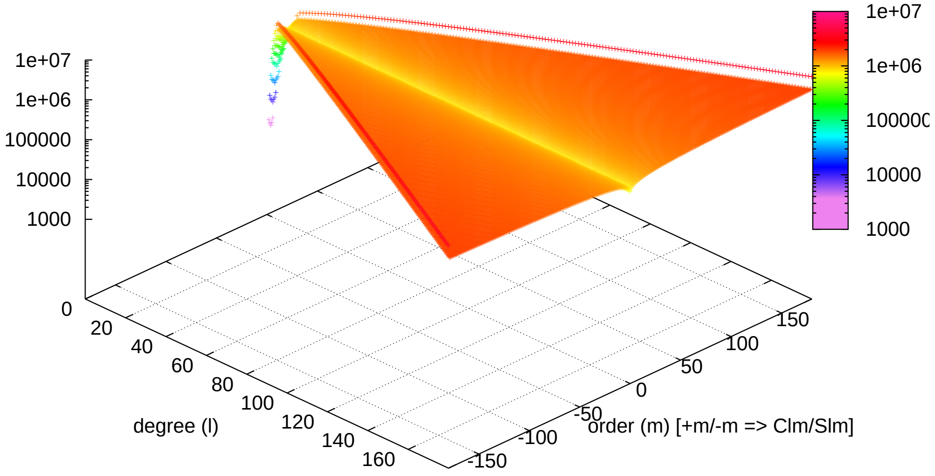


Figure 3.8: 3D plot of M_i values for all C_{lm} and S_{lm}

Using the regularization matrix M_D defined above, μ is obtained for the entire mission using the L-ribbon method. Figure (3.9) show the values of μ obtained for both M_{Kaula} and M_D. The spread of the values of μ for M_{Kaula} is much higher than that for M_D as expected. It should be noted that the values of μ for the Kaula regularization matrix is scaled by $1.5e^{-9}$ in order to compare with the values of μ obtained from M_D on the same plot.

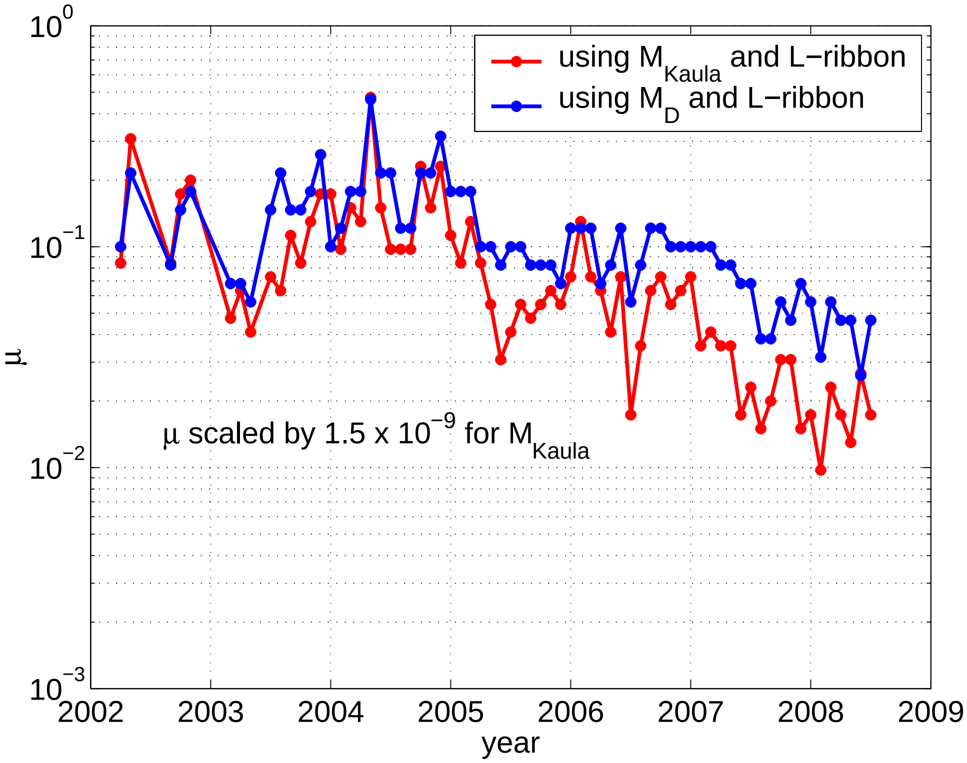


Figure 3.9: Comparison of μ values for regularization using M_{Kaula} and M_D; both using the L-ribbon method

Figure (3.10) represents a plot of the $(\sqrt{\mu M^T M})_{i,i}$ v/s degree n, for order $m = 0$, $\forall$ degrees n, and i corresponding to stokes coefficients, similar to the plot in the figure (3.7). The curves for Apr 2005 and May 2008 are plotted for both regularization matrices M_{Kaula} and M_D. As a result of the newly designed regularization matrix, we can see that the lower and the higher degrees are less constrained in the case of M_D as compared to that for M_{Kaula}. Since the M_D curve in the figure (3.10), rises steeply through the low degrees and since the spread on values μ is smaller in M_D as compared to M_{Kaula}, the constraint on the lower degrees is much less sensitive to the errors in the solution.

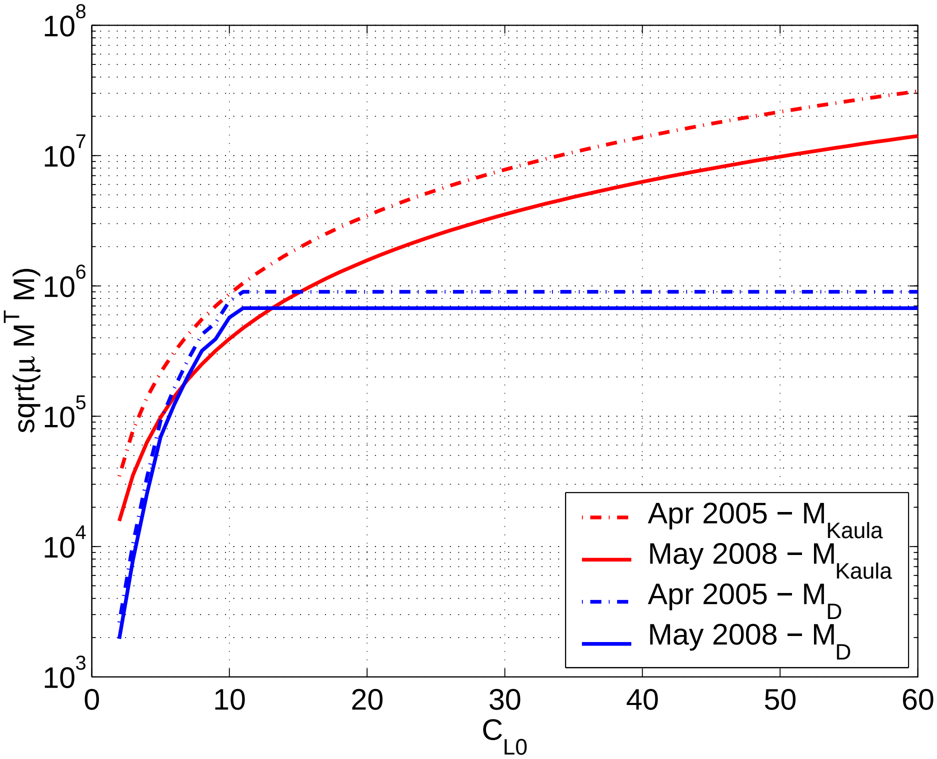


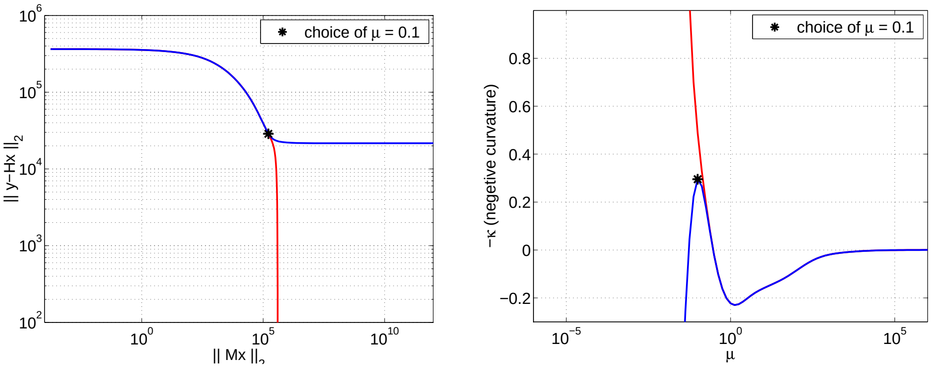
Figure 3.10: Plot of $\sqrt{\mu M^T M}$ values for C_{l0} for the solution of May 2004 and May 2008

3.5 Choice of the regularization parameter

The newly designed diagonal regularization matrix M_D is used to regularize the monthly gravity fields from GRACE in this study. As illustrated in the flowchart (2.5), the k step iterations of the parallel Lanczos bidiagonalization (LBD) algorithm is applied to the matrix $\bar{R} = RM^{-1}$, which gives us a bidiagonal matrix $C_{k+1,k}$. The iteration steps k used, varies from one parameter choice method to the other. Typically the size of the GRACE estimation problem when working with the R matrices are roughly 15000×15000 for a degree and order 120 gravity model and roughly 33000×33000 for a degree and order 180.

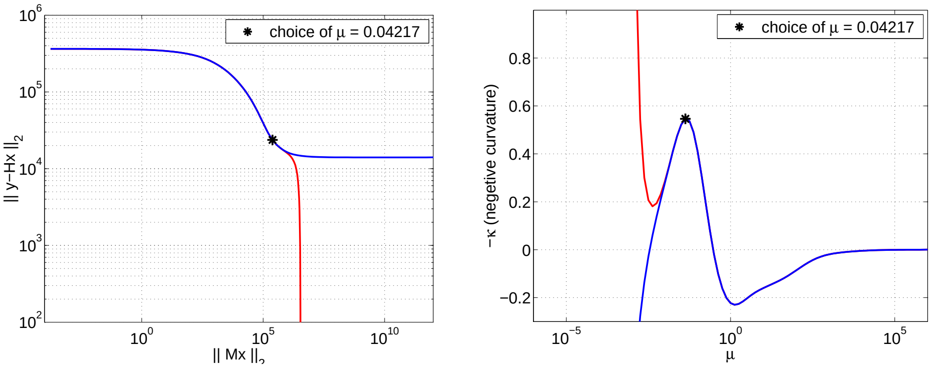
When using the L-ribbon method, a choice is made for k, using which the bidiagonal matrices are computed which are in turn used to calculate the bounds on the L-curve and its curvature. Figures (3.11) and (3.12) are the plots for these L-curve and curvature bounds for different choices of k. In these figures the left columns are the L-ribbon plots and the right columns are plots of the bounds on curvature of the respective L-curves. These bounds, represented by the red and blue curves, are plotted for different choices of k as noted. As the value of number of steps of Lanczos bidiagonalization (k) increases, the upper-bound and the lower-bound of the L-curve converges and the size of the projected problem increases. The value of μ obtained at the corner of the "L" or the maximum curvature, does not change with the increase in the LBD iteration steps k after the convergence is achieved. For $k < 100$, the corner of the L-curve is not captured in convergence of the upper-bound and the lower bound curve, making it difficult to find the optimal regularization parameter, μ, which corresponds to the corner of the L-curve. The corner of the L-curve is captured in the convergence of its two bounds with the value of around $k = 300$. $k = 300$ is thus our choice for the iteration steps of the Lanczos bidiagonalization. Any value of k greater than this value only increases the size of the projected problem thereby slowing the estimation process without any added benefit in the process of the choice of the regularization parameter. The L-ribbon method thus enables the projection of a large 16000×16000 problem on a much smaller 300×300 problem, thus making the regularization process computationally efficient. The search space for $\mu (> 0)$ and is found by trial and errors such that the corner of the L-ribbon (or maximum of the curvature of L-ribbon) lies with in the search space for μ. For the regularization matrix M_D designed in this chapter, this L-curve corner lies between 10^{-4} to 10^{4}.

If we apply the bounds on the GCV using the method described in section (2.6.4), the global minimum occurs at a point in the proximity of the μ obtained by the L-ribbon method discussed above, but the μ_{GCV} is always greater than $\mu_{L-ribbon}$. Also, as you increase the number of iteration steps k for the Lanczos bidiagonalization step, the value of μ from the bounds of the GCV obtained gets closer to that from L-ribbon method, but is still greater than that obtained by L-ribbon. This suggests a convergence problem with the GCV bounds method. In summary, using $k = 1500$ for GCV bounds gives a value of μ that is an order of magnitude greater than that obtained using



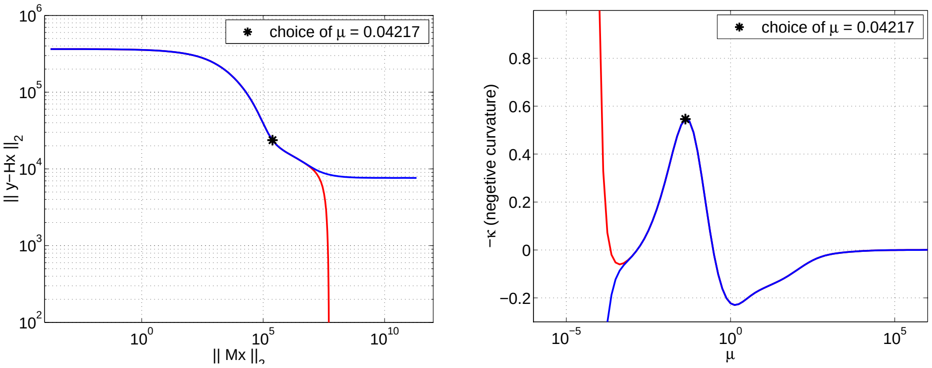
(a) L-ribbon (k=50)

(b) Curvature bounds (k=50)



(c) L-ribbon (k=100)

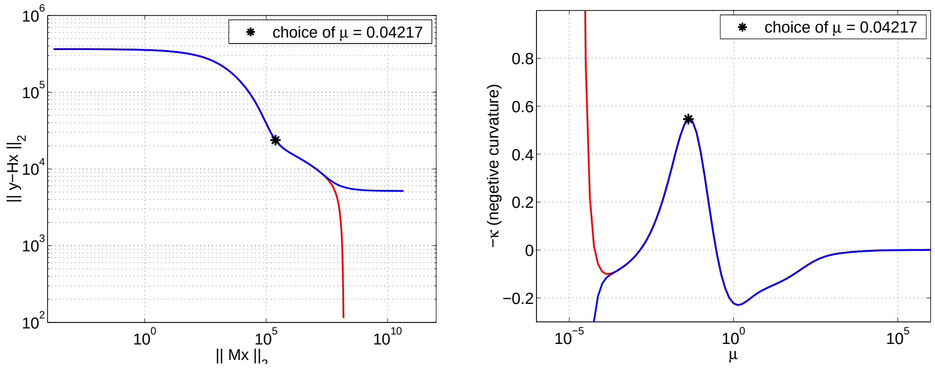
(d) Curvature bounds (k=100)



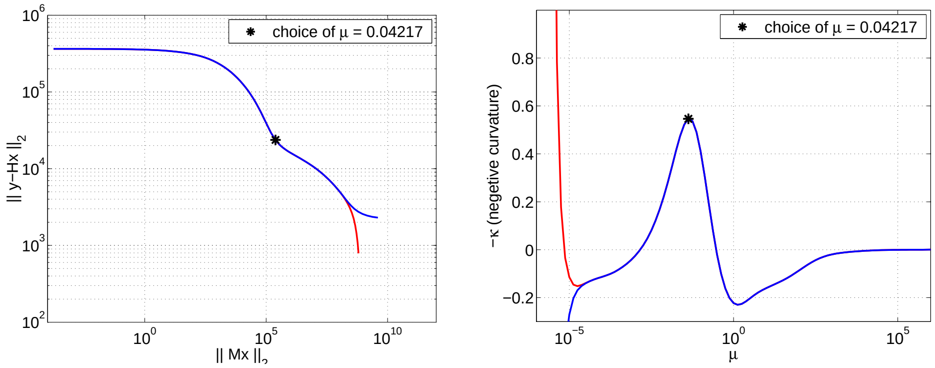
(e) L-ribbon (k=200)

(f) Curvature bounds (k=200)

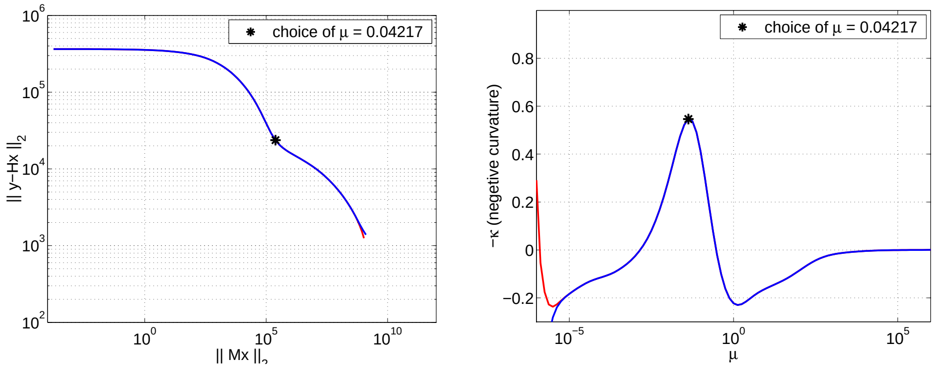
Figure 3.11: L-ribbon and Curvature-ribbon for GRACE regularization for $k = 50, 100, 200$



(a) L-ribbon (k=300) (b) Curvature bounds (k=300)



(c) L-ribbon (k=600) (d) Curvature bounds (k=600)



(e) L-ribbon (k=1000) (f) Curvature bounds (k=1000)

Figure 3.12: L-ribbon and Curvature-ribbon for GRACE regularization for $k = 300, 600, 1000$

$k = 300$ from L-ribbon and is seen in the figure (3.13). As we have discussed earlier, a larger value of μ would typically reduce more signal content in the solution than a smaller value of μ for a particular month. Also, the GCV-bounds method becomes a problem of a much larger size to solve as compared to the L-ribbon method. Hence, the value of μ used for regularization in this study is the one obtained from the L-ribbon projection method.

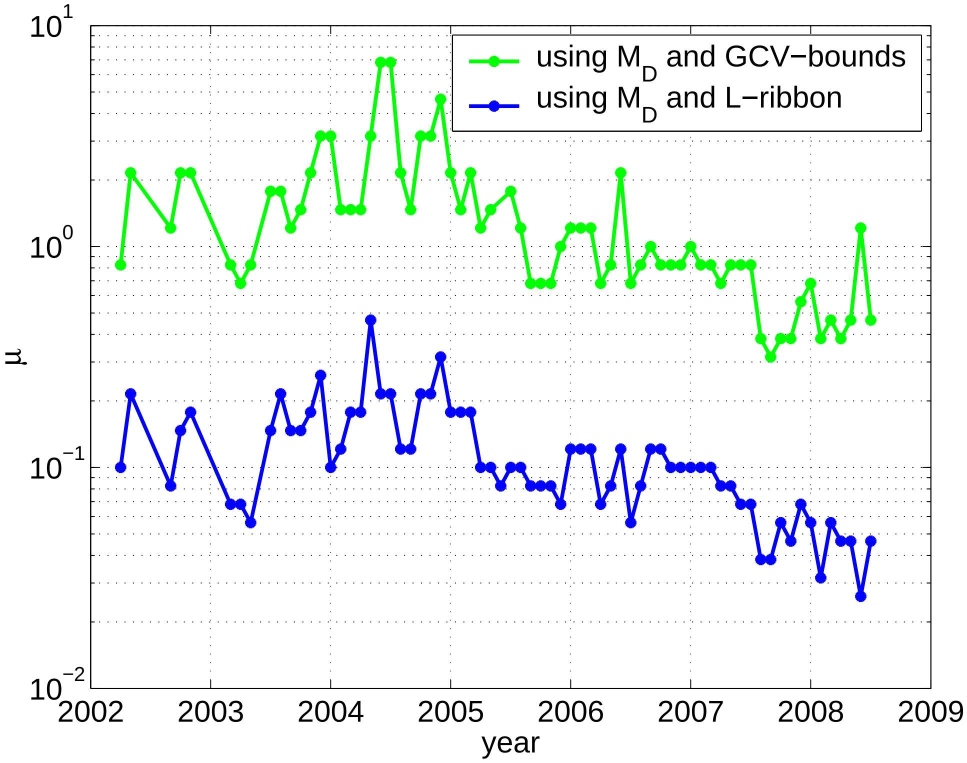


Figure 3.13: Comparison of μ values for regularization using L-ribbon and GCV-bounds methods for M_D

As noted in the literature [33] [24], the L-curve typically gives results that are over smoothed, which is the result of a higher than optimal value of μ. Hence, we will use two regularization parameters to compute two different sets of solution to check for the possible over-smoothing effect of the L-ribbon method. The regularization parameters used are:

1. μ_{lr} that is obtained from the L-ribbon method
2. $\mu_s = 0.4\mu_{lr}$, by scaling down the L-ribbon μ_{lr} to counter the effect of possible over-smoothing with L-ribbon. This scale factor of μ was chosen by visual inspection of the maps of mass flux and the maps of the residuals in order to obtain a balance between the errors-strips and loss of signal.

The two sets of solutions obtained using these two regularization parameters will be named REG_{LR} and REG_S respectively, for the remainder of the discussion and the results are discussed in the next chapter.

Chapter 4

Results

The choices made for the regularization process for GRACE are discussed in chapter (3). Using these choices for the regularization matrix M_D and the regularization parameters from lribbon, μ_{lr} and scaled L-ribbon, μ_s, the GRACE solutions are regularized for the entire mission starting April 2002 to July 2008. the regularized solutions are solved to degree and order 120. The two time series of solutions obtained are referred to as REG_{LR} and REG_S corresponding to the regularization parameters μ_{lc} and μ_s respectively. These solutions will be compared to the unconstrained GRACE release 4 (RL04) solutions which are up to degree and order 60. In the months of Jan 2003 and Feb 2003, there were problems with the attitude data in one of the satellites, resulting in uncharacteristically large errors in the solutions. The singular values in the R matrix for the months of Aug 2002, Dec 2002, Jan 2003 and Feb 2003 are such that for the regularization matrix M_D, the L-ribbon for these months does not produce a corner in the search space for μ. These months will require special handling with the redesign of the regularization matrix M to handle the errors in these months, and have thus been ignored in these study. The months of June 2002, July 2002 and July 2003, there were problems in the accelerometer data in one of the satellites and hence have no solutions. The solutions obtained are analyzed for signal and noise content in this chapter. The evaluation of a gravity field is very subjective and there are no definitive tests to determine if one field is better than another. Instead, there are a series of comparisons that we make to draw some general conclusions about the regularized gravity field.

4.1 Variability

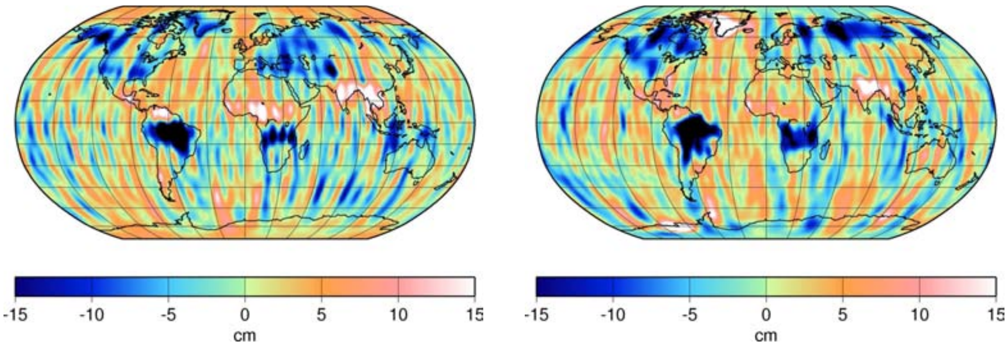
The first assessment of the GRACE solution is done by looking at the map of the monthly variability. Monthly variability is defined as the deviation in the gravity field estimates with respect to a long term mean as discussed in chapter (3). An arithmetic average of multiple years of monthly GRACE gravity field models are computed for each of the separate time series (RL04, REG_{LR} and REG_S). We define this long term mean in this study using 60 months from August 2003 to July 2008. The monthly variability with respect to this mean is then plotted as a map of the mass flux as equivalent water heights for all the months [57]. Figures (4.1) and Figures (4.2) are the maps of the monthly variabilities of mass flux in the units of centimeter (cm) of equivalent water heights and a comparison is made between the three different solutions (RL04, REG_{LR} and REG_S). In both the figures, a Gaussian smoothing of 350km has been applied to the RL04 maps while no smoothing is applied to the maps of regularized (REG_{LR} and REG_S) fields. The figure (4.1) compares the variability maps for the months of Apr 2005 and May 2008 that have relatively low error-stripes in the RL04 fields to begin with.

Both the regularized solutions have lower magnitude of noise/stripes in the solutions as compared to RL04. But the improvement in the solutions is clearly evident when we compare the two months with relatively high level of errors. Figure (4.2) compares variability maps for such months with high errors like Jan 2004 and Sep 2004. It can be observed that the noise in the regularized solutions is vastly reduced and the quality of the regularized solutions is much more uniform over the mission as compared to the RL04 solutions. As expected, the REG$_S$ solutions are more noisier than REG$_{LR}$ because the value of μ used to compute the REG$_S$ solution is smaller than that used for REG$_{LR}$ by a factor of 0.4. In addition to looking at the variability maps, signal analysis is performed on the solutions to make sure that the signal has not been attenuated by the regularization process.

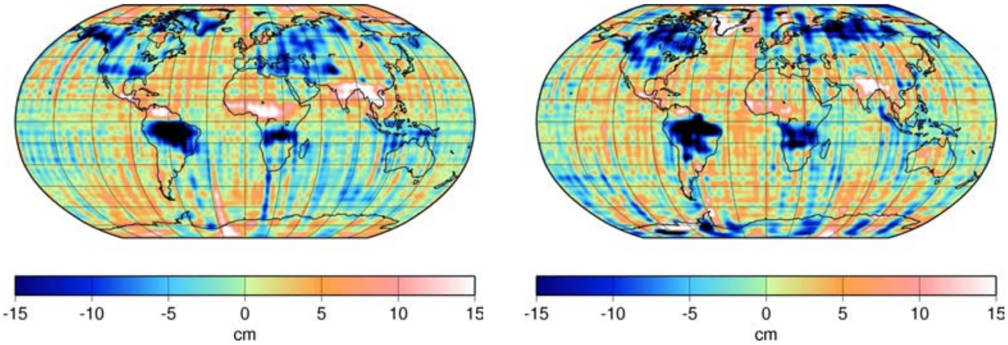
4.2 Degree variance

The other assessment of the quality of the field is to plot the degree variance, degree error variance and the degree difference variance. Given the set of normalized Stokes coefficients, $(\bar{C}_{lm}, \bar{S}_{lm})$ and their respective standard deviations $(\delta\bar{C}_{lm}, \delta\bar{S}_{lm})$, the degree variance (DV) is calculated as follows:

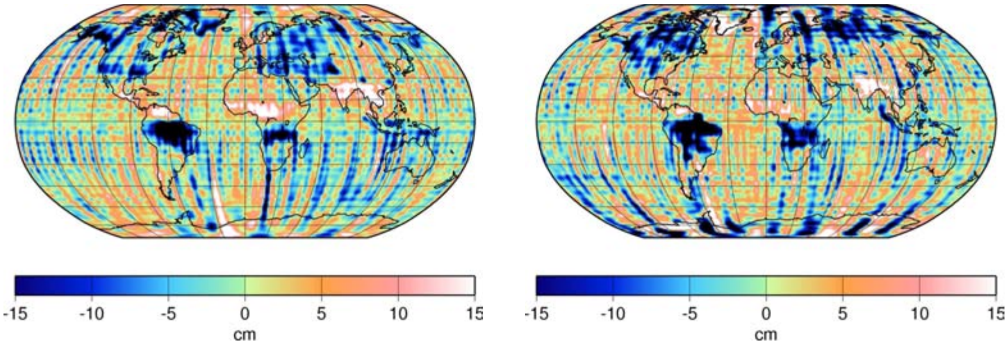
$$DV_l = \sqrt{\sum_{m=1}^{l}(\bar{C}_{lm}^2 + \bar{S}_{lm}^2)} \tag{4.1}$$



(a) Apr 2005 - RL04 - 350 km Smoothing (b) May 2008 - RL04 - 350 km Smoothing

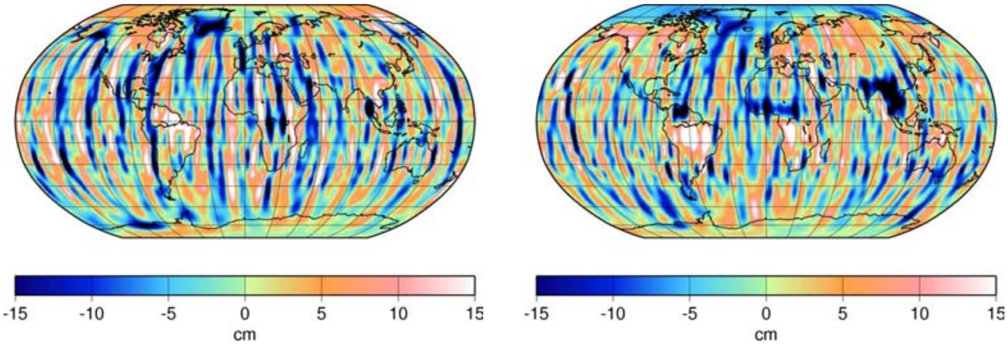


(c) Apr 2005- REG$_{LR}$ - No Smoothing (d) May 2008 - REG$_{LR}$ - No Smoothing

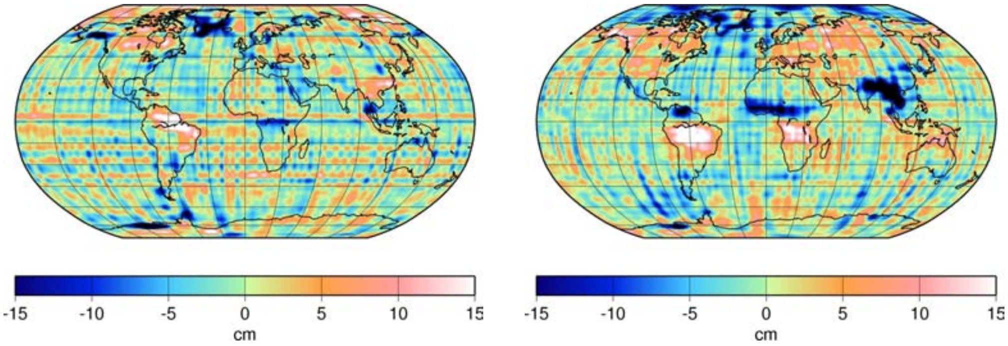


(e) Apr 2005 - REG$_S$ - No Smoothing (f) May 2008 - REG$_S$ - No Smoothing

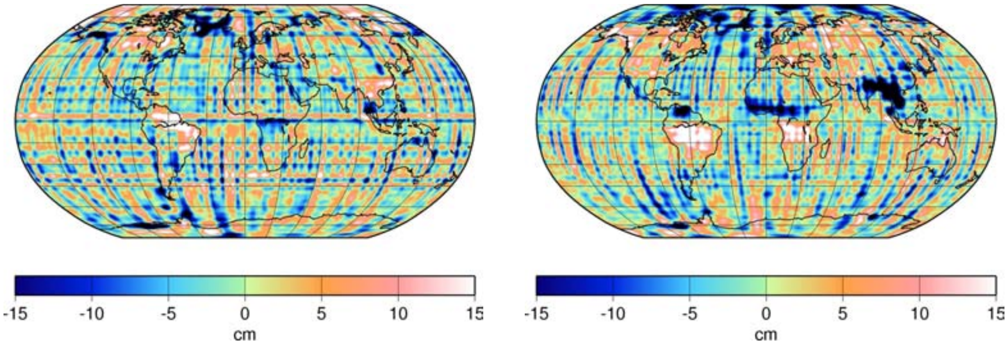
Figure 4.1: Variability (cm of equivalent water height)



(a) Jan 2004 - RL04 - 350 km Smoothing (b) Sep 2004 - RL04 - 350 km Smoothing



(c) Jan 2004- REG$_{LR}$ - No Smoothing (d) Sep 2004 - REG$_{LR}$ - No Smoothing



(e) Jan 2004 - REG$_S$ - No Smoothing (f) Sep 2004 - REG$_S$ - No Smoothing

Figure 4.2: Variability (cm of equivalent water height)

where l and m are the respective degree and order for the geopotential coefficients. Similarly the degree error variance (DVE) is computed as

$$DEV_l = \sqrt{\sum_{m=1}^{l}(\delta\bar{C}_{lm}^2 + \delta\bar{S}_{lm}^2)} \tag{4.2}$$

The degree difference variance (DDV) is used to compare the difference between the estimates of the two fields as follows:

$$DDV_l = \sqrt{\sum_{m=1}^{l}(\Delta\bar{C}_{lm}^2 + \Delta\bar{S}_{lm}^2)} \tag{4.3}$$

The DV, DEV and DDV values are all scaled by the Earths radius (6378136.3 km) in order to express the results in terms of geoid height. The figure (4.3) is the plot of RMS of the DDV and DEV for the three gravity field time series. DDV is calculated with respect to the population mean using in the section (4.1. The DDV curve for the regularized time series is very consistent with the RL04 time series upto degree 10 after which they deviate from each other. The degree amplitude curves for the models of hydrology, oceanography and atmospheric signal in [57] are decreasing through degree 40, unlike the DDV curve of the RL04 variability which increases after degree 20. The decreasing nature of the DDV curves of the regularized solutions is consistent with the degree amplitude curves of the models. The REG$_S$ has more power as compared to the REG$_{LR}$ solutions which is because of the presence of more errors in the REG$_S$ fields but there is a possibility that some of this energy could be real signal.

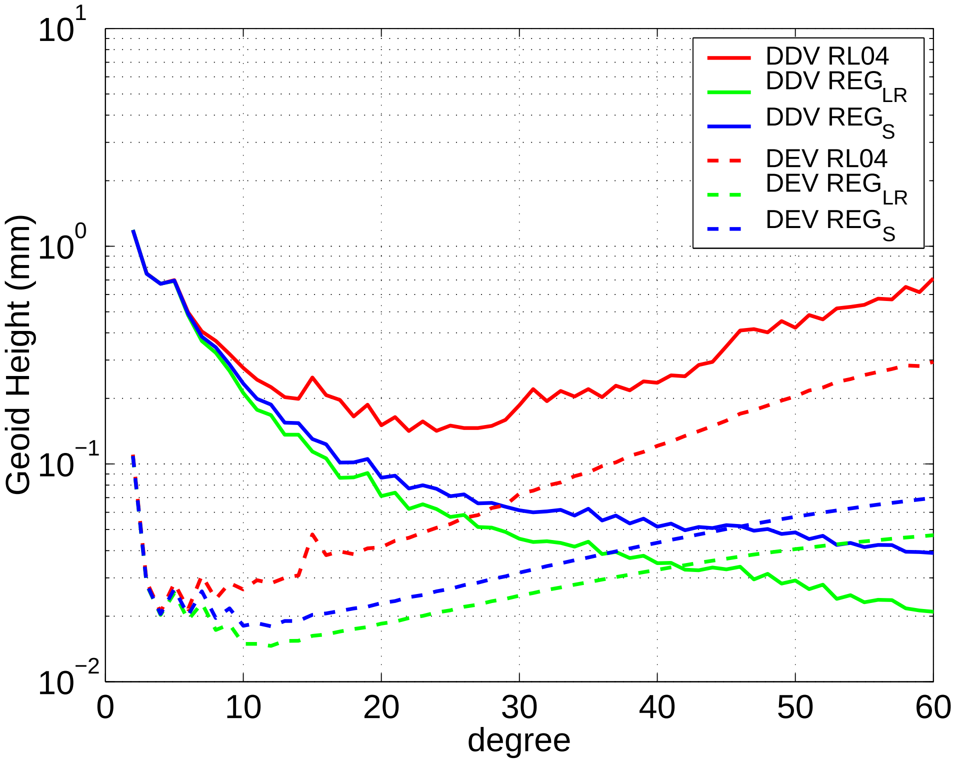


Figure 4.3: Degree difference variance with respect to the population mean and error variance

The DDV of the regularized fields generated by the group at *Universitt Bonn* [39] is plotted along with the DDV for the RL04 and regularized solutions in the figure (4.4). The ITG time variable gravity field is parametrized by quadratic splines in time domain and up degree 40 using stokes coefficient in the space domain. The quadratic splines have a nodal point distance of half a month (about 15 days) to each nodal point there belongs a set of spherical harmonic coefficients. In the estimation process the variations are filtered by applying a regularization matrix for each set of spherical harmonic coefficients. The regularization matrices were chosen by analyzing hydrological models and have a Kaula type form. For each regularization matrix the regularization parameter was determined by the variance component estimation method. The reader is referred to [54] for more details about this time-series. The time variable ITG solutions appear to have less energy at the lower degrees in the degree variance as compared to the GRACE fields, both RL04 and regularized, which probably results in lower variability signal in the gravity fields.

4.3 Post-fit residual analysis

Post-fit analysis is a way to make sure that all the geophysical signal in the observations is captured in the gravity field estimate. The estimate ($\hat{x}$) of the Stokes coefficients and other model parameters obtained are fitted to the observations to get the residuals ($y - H\hat{x}$). These residuals are in the units of range-rate (nm/s). One cycle per revolution signal is removed from these residuals and CRN filtering with 10 mHz bandwidth is then applied to generate

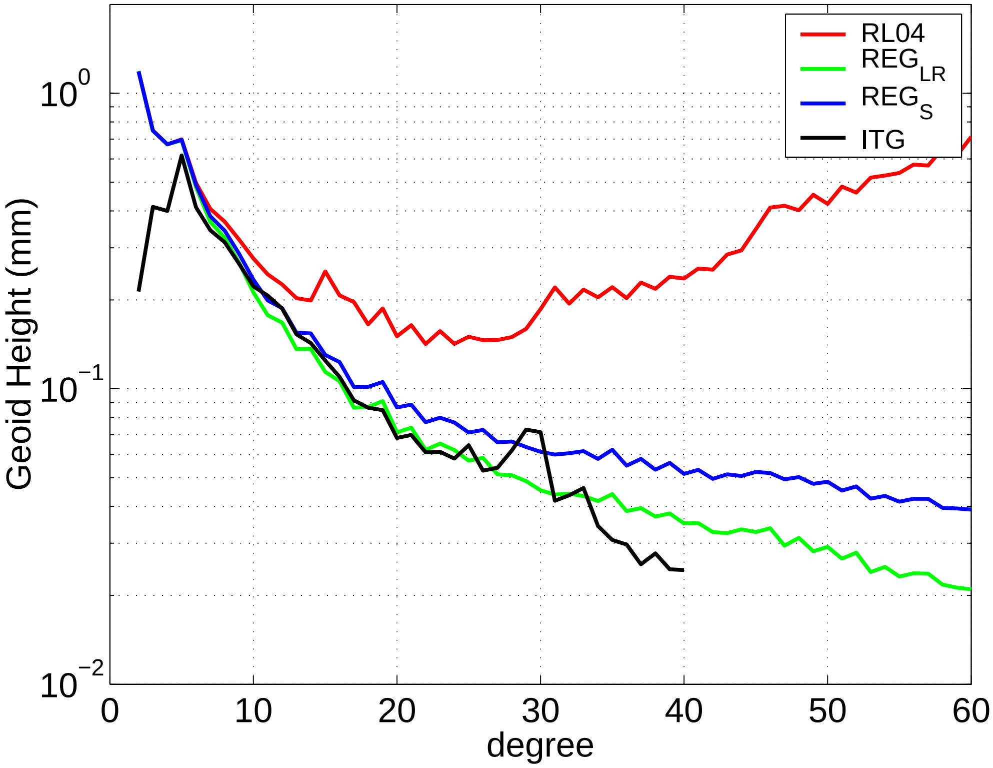
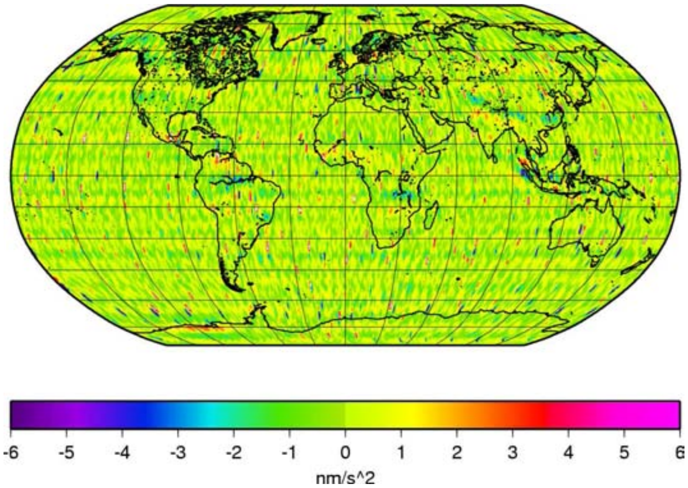


Figure 4.4: Degree difference variance w.r.t. the population mean

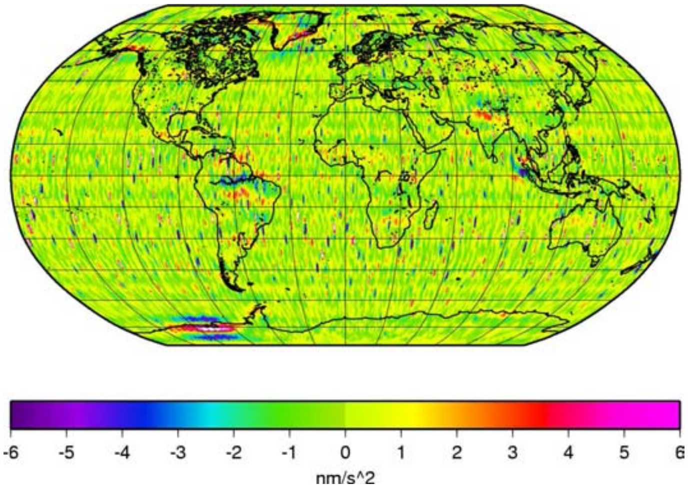
differentiated residuals in the units of range accelerations (nm/s^2). These residuals are then geographically mapped using the state information of the GRACE satellites. These maps should not show any geophysical correlation if all the time variable signal has been captured by the solution.

The RL04 solution is used in the science community with certain post processing like Gaussian smoothing or "de-striping". But both these methods of preprocessing will attenuate the signal while removing the stripe-errors in the GRACE solutions. To put this loss of signal in perspective with the possible loss of signal in the regularization process, the Gaussian smoothed solution at 350km, is fitted to the observation data as described above. Figure (4.5) and and figure (4.6) are the maps of the posit-fit residual computed using the Gaussian smoothed solution and RL04 respectively. It can be observed that while the residual maps for RL04 do not show any geo-physical characteristics, the residual map of the Gaussian smoothed solution does show a lot of geophysical signal in the residuals in the areas like Alaska, Antarctica, Himalayas, the great Sumatra earthquake of 2004, Amazon etc. This suggests that when the Gaussian smoothing of 350km applied to the solution, there is significant loss of signal when it removes the errors in the solutions. It should be noted here that when the Gaussian smoothing is performed over a river basin, the possible attenuated signal is restored by scaling the attenuated signal by a factor dependent on the area of the basin. This factor usually replaces some of the signal lost due to smoothing over the basins.

The same post-fit residual analysis is performed with the regularized

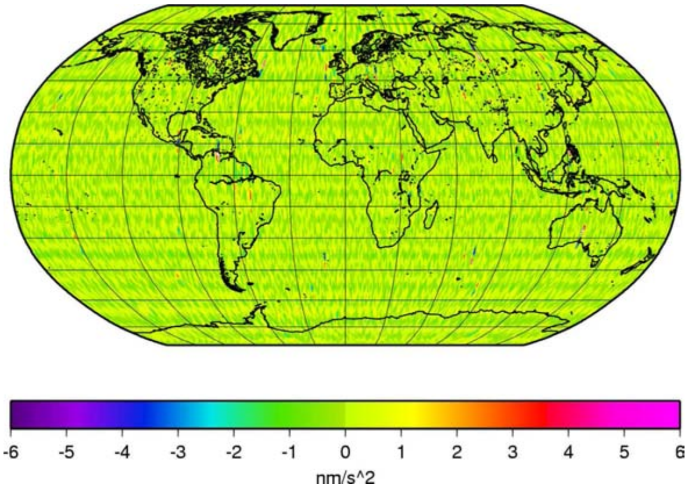


(a) Apr 2005

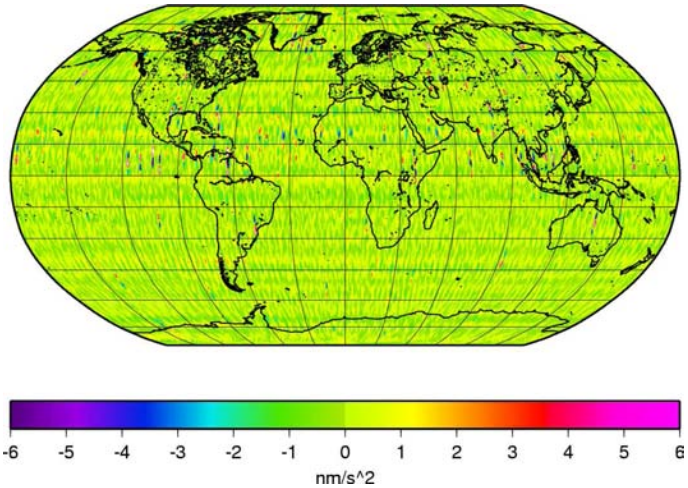


(b) May 2008

Figure 4.5: Residual after fitting the 350 km Gaussian smoothed solutions to the data in the units of range acceleration (nm/s^2)

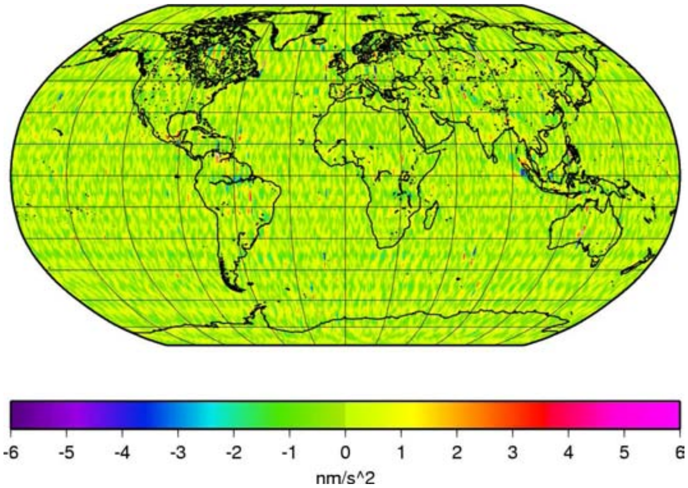


(a) Apr 2005

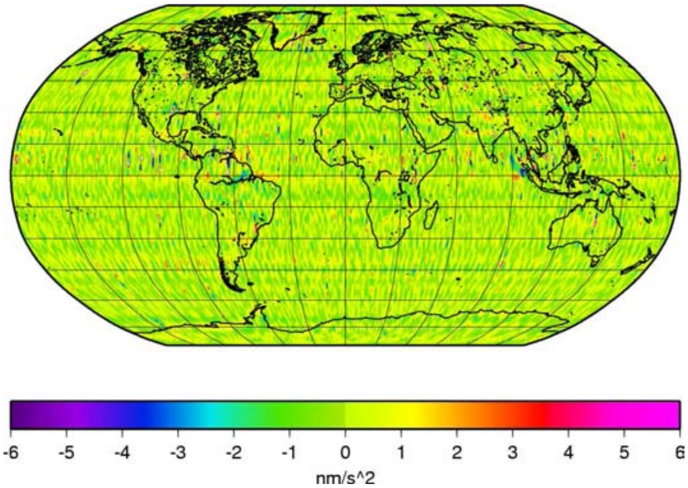


(b) May 2008

Figure 4.6: Residual after fitting the RL04 solutions to the data in the units of range acceleration (nm/s^2)

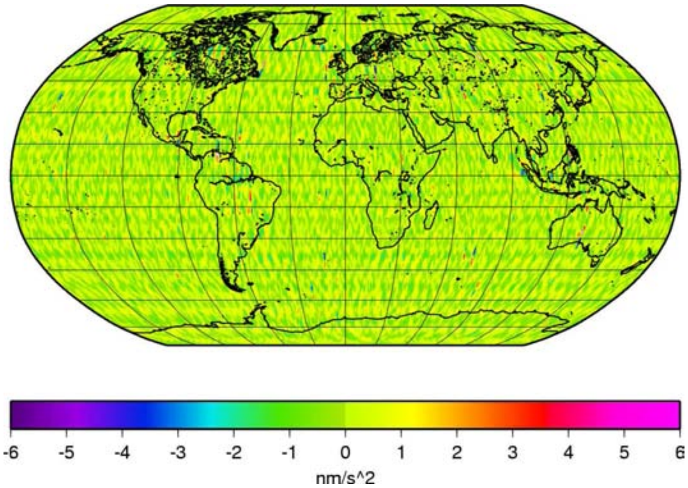


(a) Apr 2005

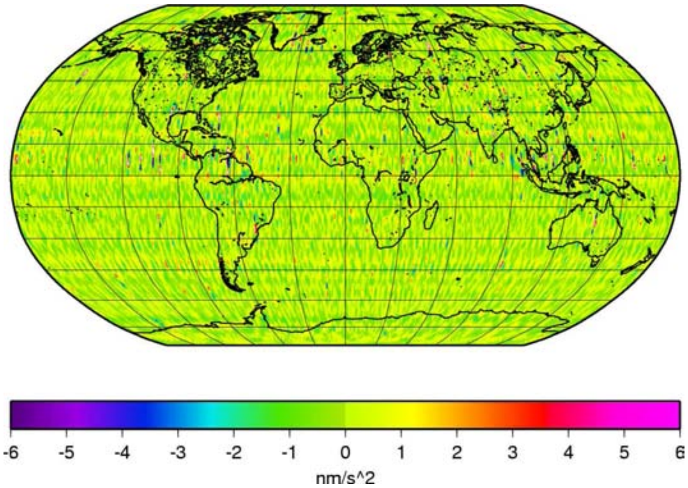


(b) May 2008

Figure 4.7: Residual after fitting the REG_{lr} solutions to the data in the units of range acceleration (nm/s^2)



(a) Apr 2005



(b) May 2008

Figure 4.8: Residual after fitting the REG_S solutions to the data in the units of range acceleration (nm/s^2)

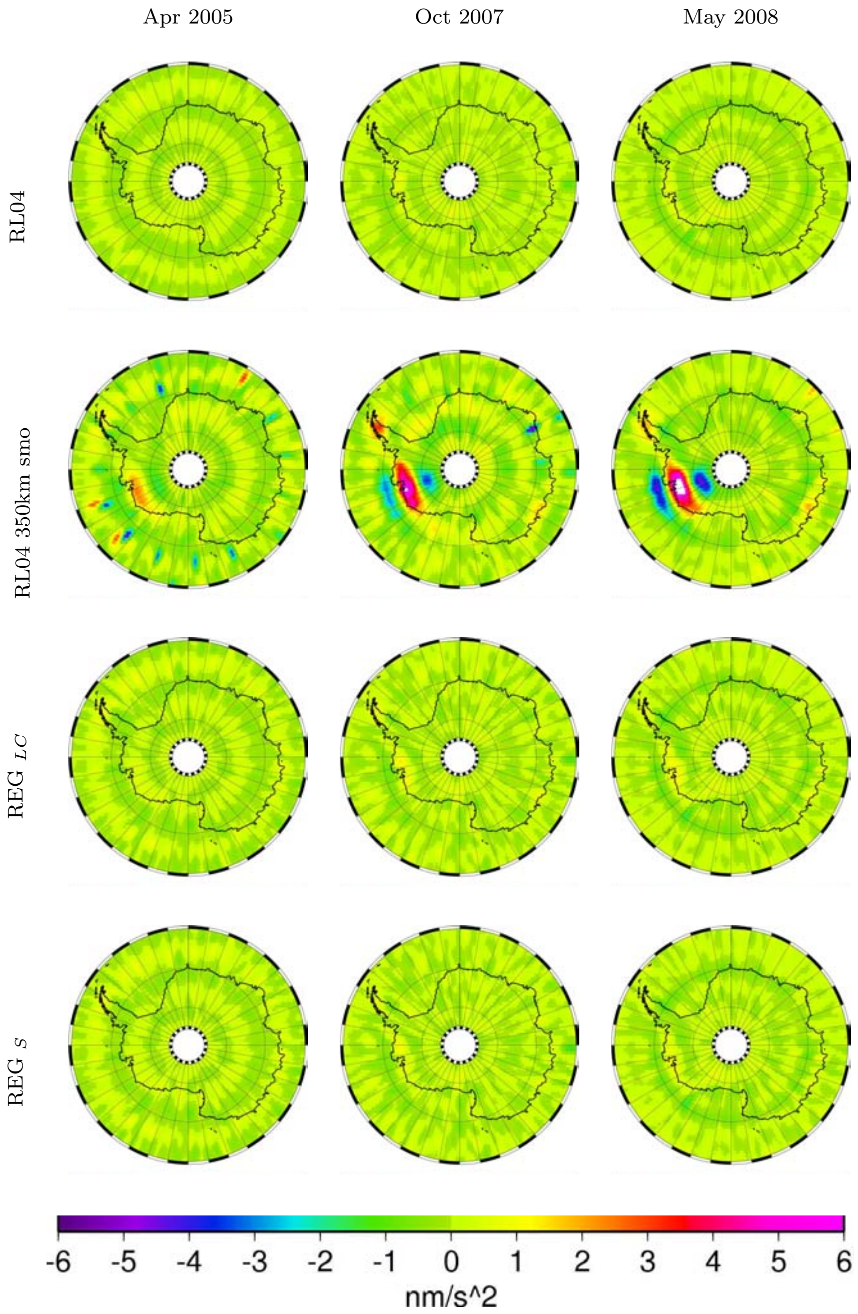


Figure 4.9: Post-fit residuals over Antarctica (nm/s^2)

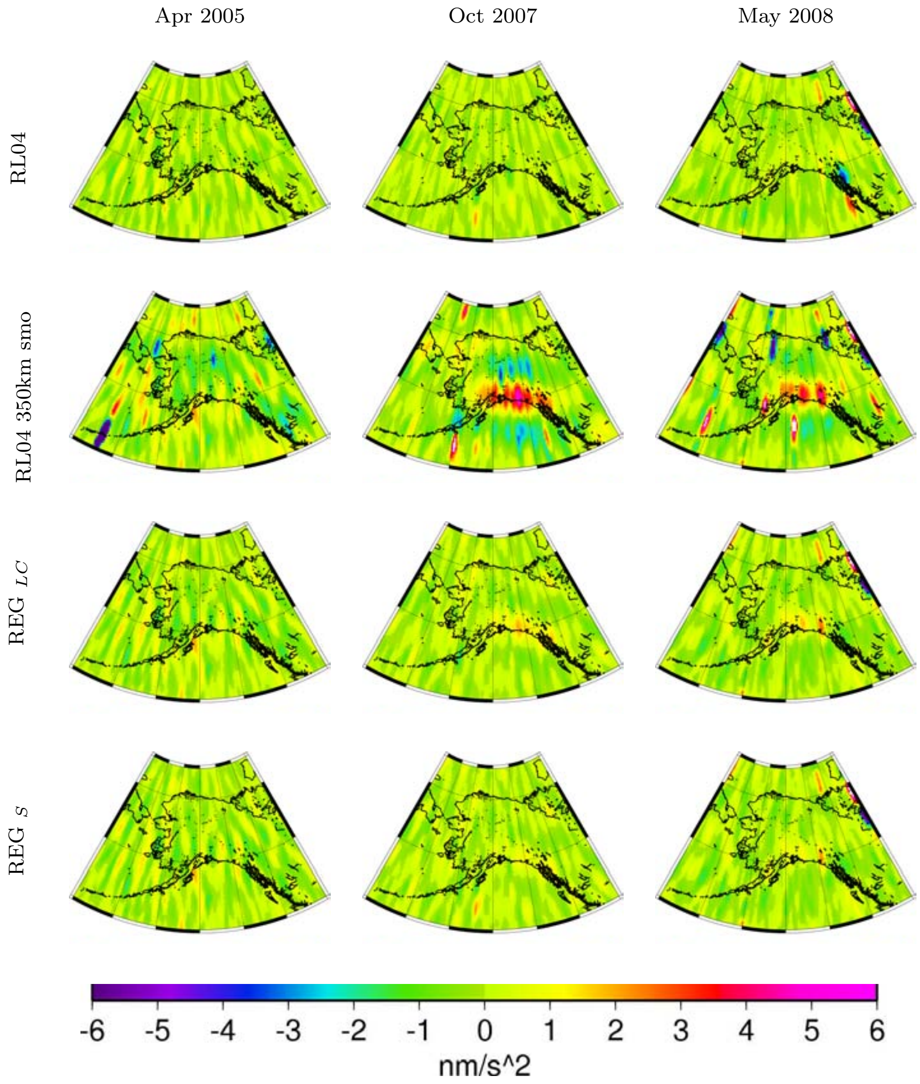


Figure 4.10: Post-fit residuals over Alaska (nm/s^2)

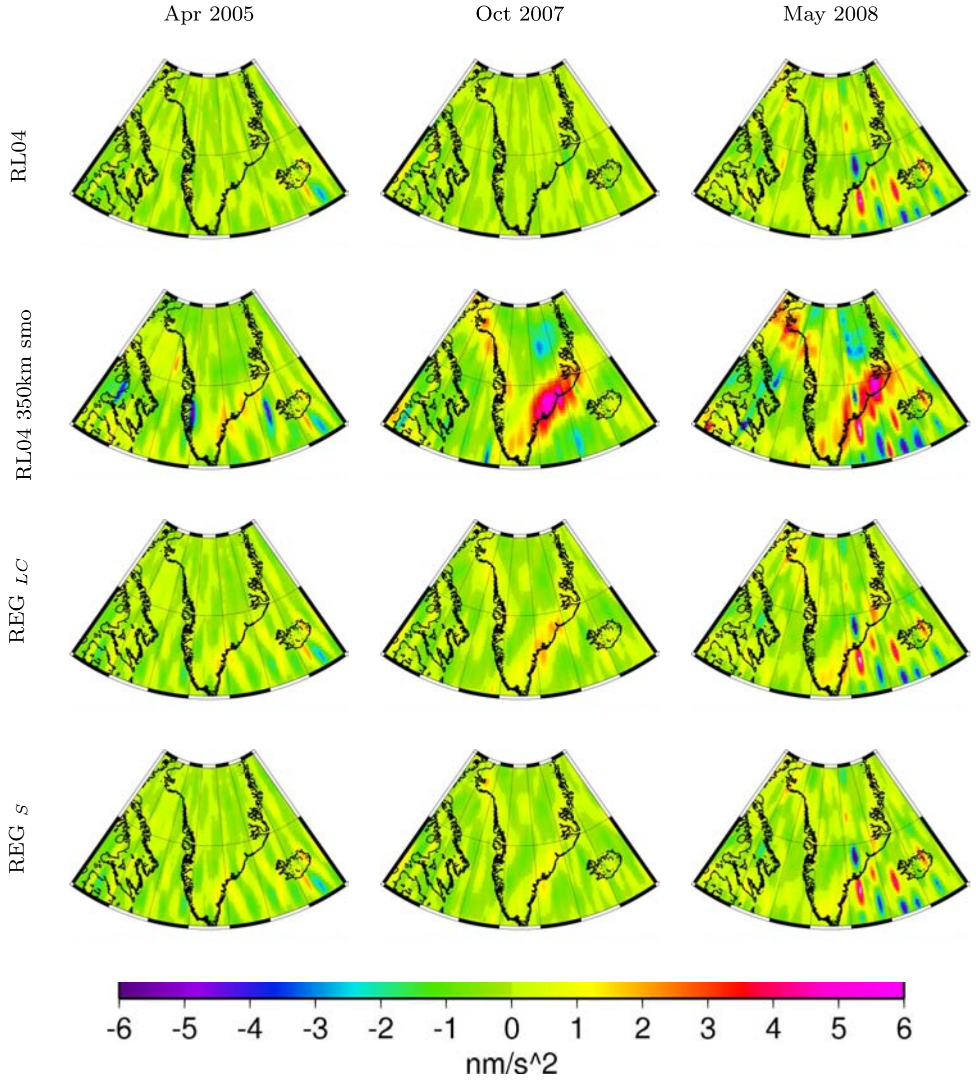


Figure 4.11: Post-fit residuals over Greenland (nm/s^2)

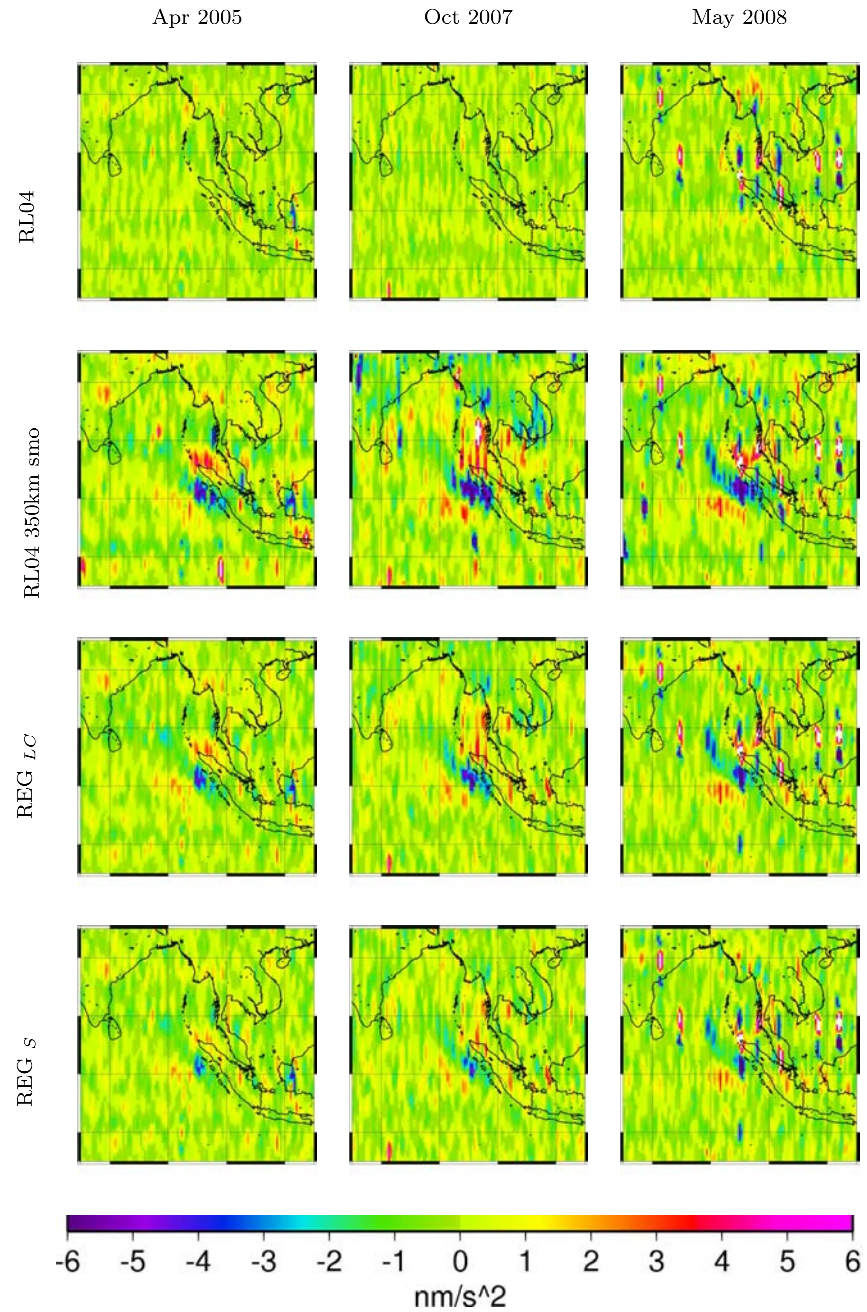


Figure 4.12: Post-fit residuals over Sumatra earthquake region (nm/s^2)

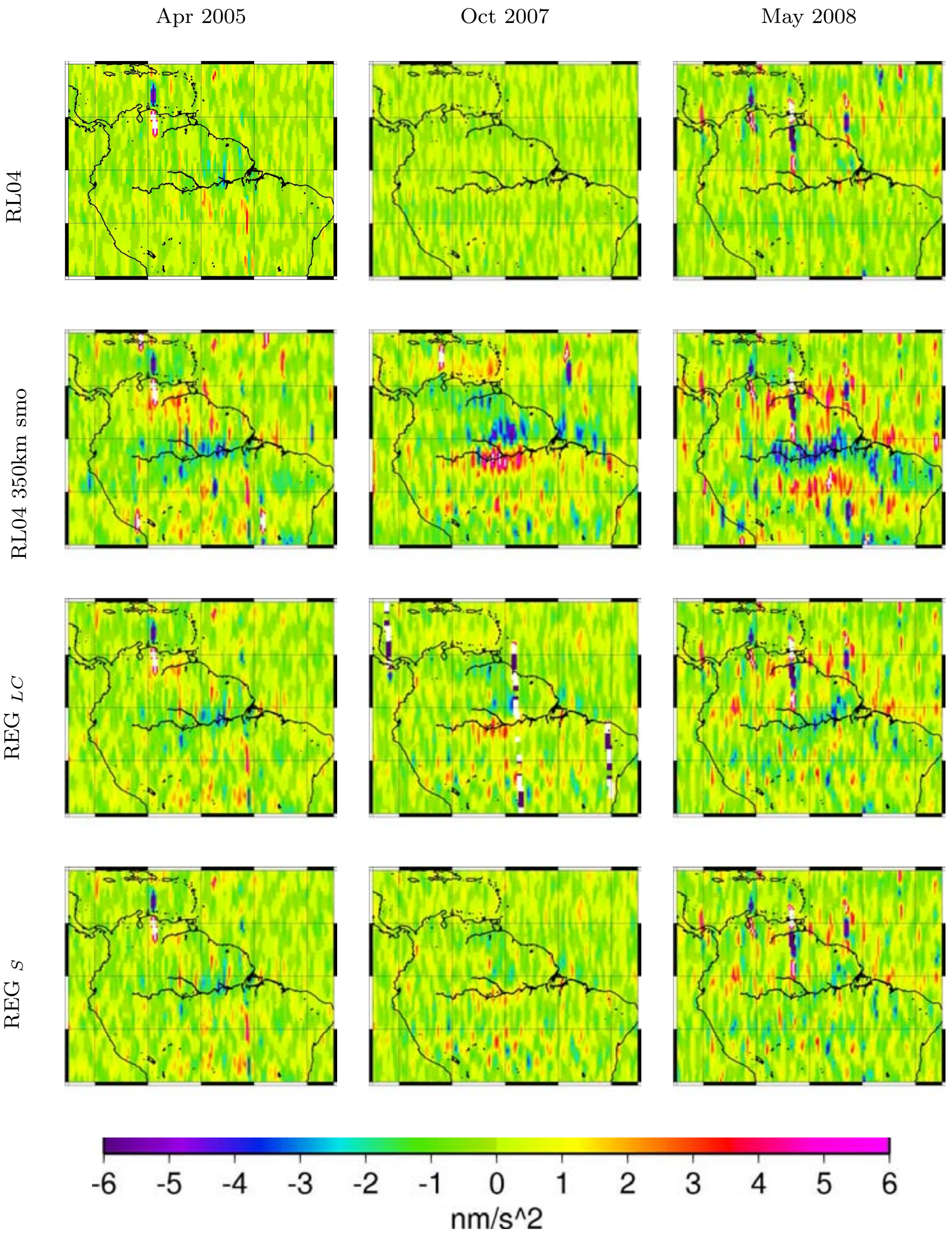


Figure 4.13: Post-fit residuals over Amazon (nm/s^2)

solutions and the maps are shown in the figures (4.7) and (4.8). At the global scale, there is little or no geophysical signal visible in the residuals of the regularized solutions, as compared to the signal visible in the residuals of the Gaussian smoothed solutions. The residuals of all the solutions are compared in the high amplitude and slope regions of Antarctica (figure 4.9), Alaska (figure 4.10), Greenland (figure 4.11), Sumatra Earthquake (figure 4.12) and the Amazon (figure 4.13) for the three months, Apr 2005, Oct 2007 and May 2008. There is signal in the residuals in the regions with high geophysical signal in a very small area, like the great Sumatra earthquake of 2004, for both the regularized solutions. There is slight evidence of geophysical signal in the residual of REG$_{LR}$ solutions in the regions with heavy mass loss due to ice melting, like Greenland, Antarctica and Alaska but the residuals are extremely small compared to the post-processing method. There is no visible signal in the residuals of the REG$_S$ solutions. Thus, the REG$_S$ solution captures all the variability signals in the observations, but fails to capture the Sumatra earthquake signal.

4.4 Correlation between RL04 and regularized solutions

If $\hat{c}^i_{lm}(t)$ and $\hat{s}^i_{lm}(t)$ denote the spherical harmonics co-efficients difference with respect to the nominal gravity field at time t for the time series i, then the degree variance of this update is given by [1],

$$\sigma_l^{i2}(t) = \sum_{m=0}^{l} \left[{\hat{c}_{lm}^{i}}^{2}(t) + {\hat{s}_{lm}^{i}}^{2}(t) \right] \tag{4.4}$$

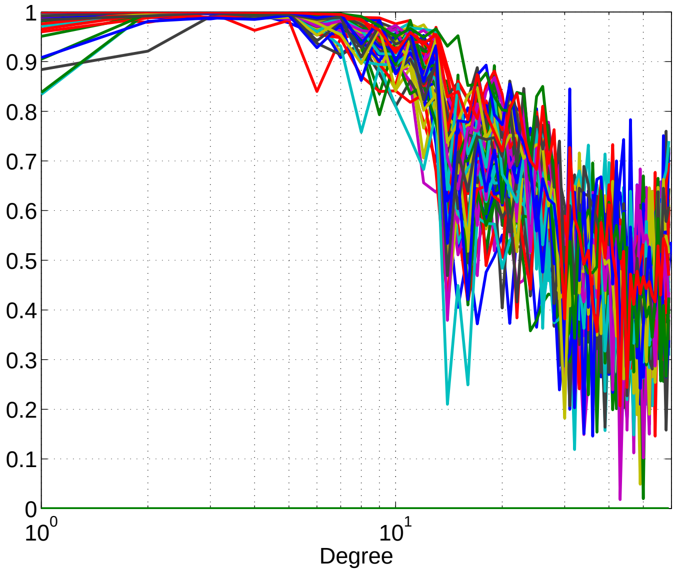
And the degree cross-correlation of one series i with respect to the other series j at time (t) is given by,

$$\rho_l^{ij}(t) = \frac{1}{\sigma_l^i \sigma_l^j} \times \sum_{m=0}^{l} \left[\hat{c}_{lm}^{i}(t) \times \hat{c}_{lm}^{j}(t) + \hat{s}_{lm}^{i}(t) \times \hat{s}_{lm}^{j}(t) \right] \tag{4.5}$$

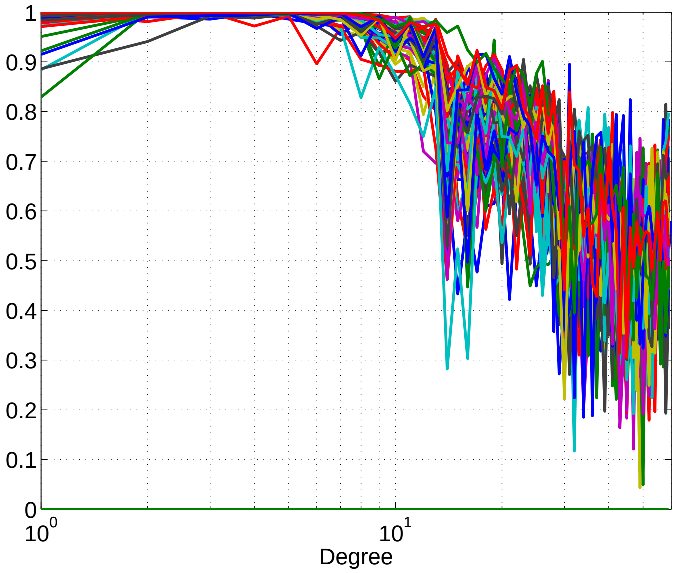
Thus for each monthly gravity field we have one degree correlation curve. These curves are plotted in the figure (4.14a) and figure (4.14b) where each curve represents a monthly degree cross-correlation between each of the regularized and RL04 solutions. Figure (4.15) is the mean of all the curves and represents the degree cross-correlation between each of the regularized series and RL04 in the mean sense over the mission. The cross-correlation is above 0.8 up to degree 14 between them. This suggests that all of the signal in the lower degrees is captured while removing the errors in the high degrees. The figure also shows that REG_S is correlated with RL04 up to degree 10 with the a correlation of 0.9 or greater. The drop in the correlations at degree 15 is expected since we see large errors in the degree 15 coefficients in the RL04 estimates due to resonance problems, which is not seen in the regularized solutions.

4.5 Error and signal over the ocean

The amplitude of monthly variability signal over the ocean is small compared to that on the land. Hence, it is easy to separate stripes over the



(a) REG_{LR} w/ RL04



(b) REG_{S} w/ RL04

Figure 4.14: Cross correlation of the regularized series with respect to RL04. Each curve represents a monthly solution

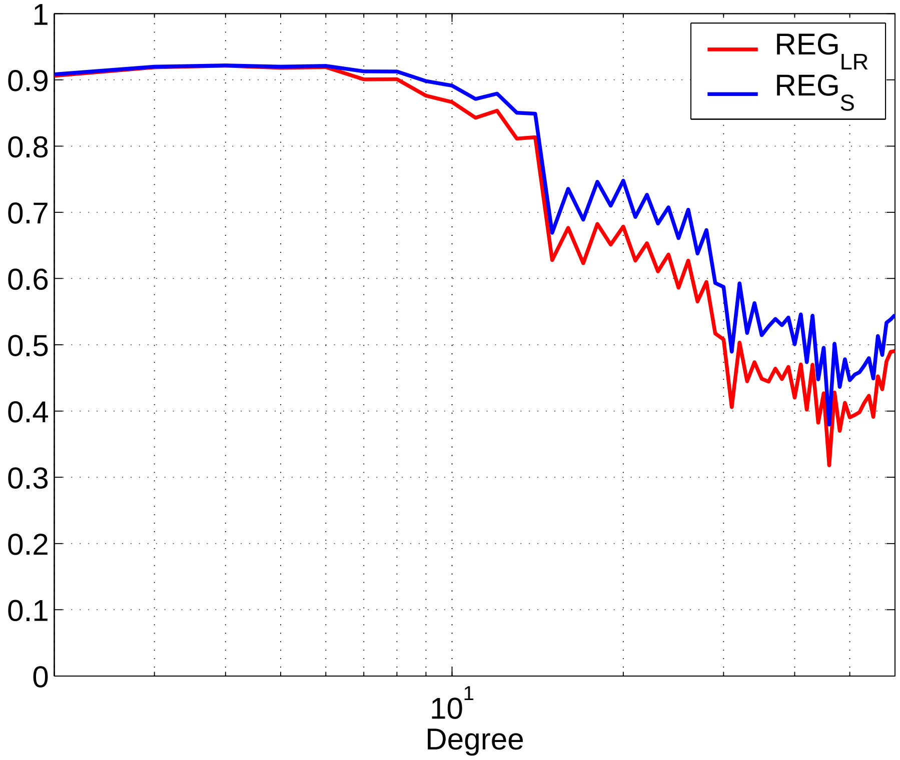
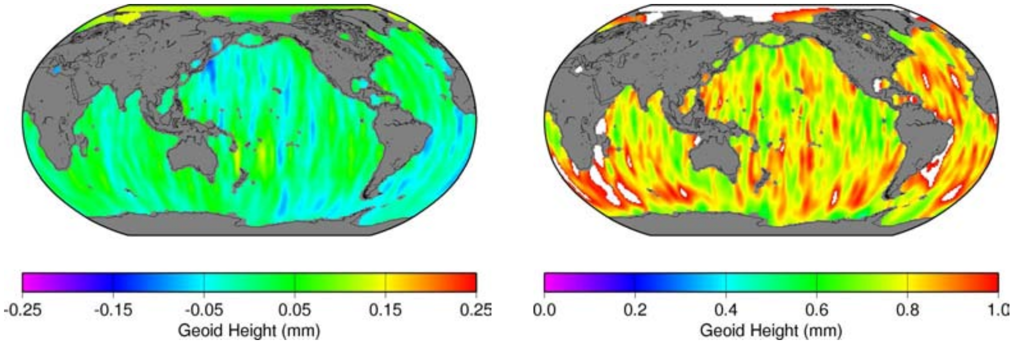


Figure 4.15: Cross correlation of the regularized series with respect to RL04 calculated as a mean over the time series

ocean. Sinusoids can fit most of the time variable signal over the ocean in the Earth's gravity field time series. An 8 parameter model with a slope, an intercept, and sinusoids with 3 periods is fitted to the solution time-series and a residual with respect to this fit is computed. The three periods used are the annual, the semi-annual and 161 day alias period arising from the errors in S_2 tide [40]. These monthly residuals with respect to the 8 parameter fit contains all of the strips and the signal that is not captured by the 8 parameter fit. Over the ocean, these residuals contain mostly errors. Thus a mean and RMS of these residuals over the global ocean is computed for both the time series. Figure (4.16) shows that the mean of these residuals with respect to the 8 parameter fit is comparable for both RL04 series and regularized series. This is because most of these errors are systematic and cancel out when we take its long term mean. But the RMS of these residuals show that the RL04 solution has a very high RMS as expected. But the RMS of the residuals for the regularized time series show a pattern. They are low at the equator and high at the high latitudes. The signal in the southern sea, south west of Australia, shows up in the regularized solutions but is masked by the errors in the RL04 solutions.

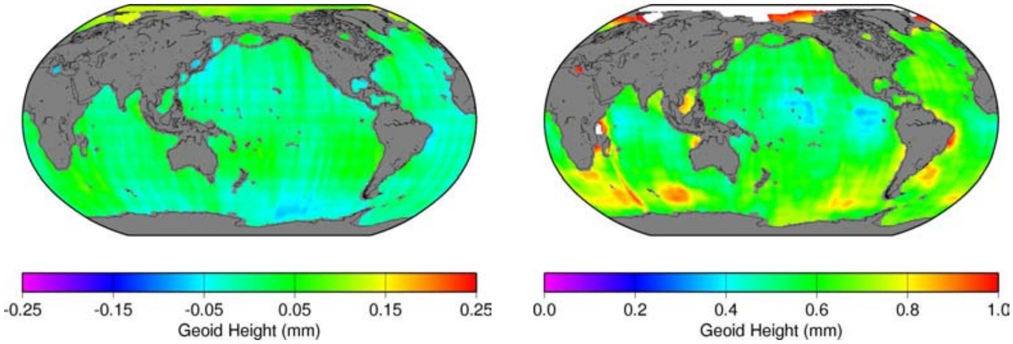
4.6 Basin average analysis

The time variable gravity changes are caused by a combination of post-glacial rebound, fluctuations in atmospheric mass, and the redistribution of water, snow and ice on land and in the ocean. The data from GRACE dramat-



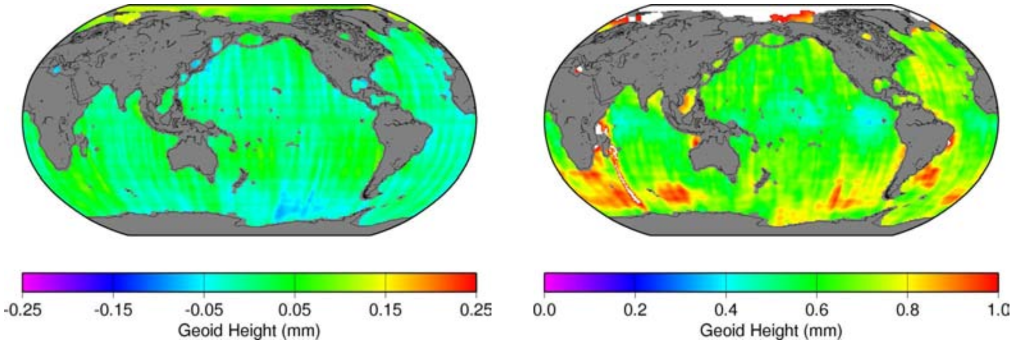
(a) Average - RL04 (300 km Smoothing)

(b) RMS - RL04 (300 km Smoothing)



(c) Average - REG_{LR} (No Smoothing)

(d) RMS - REG_{LR} (No Smoothing)



(e) Average - REG_S (No Smoothing)

(f) RMS - REG_S (No Smoothing)

Figure 4.16: Statistics of the residuals over the ocean with respect to the seasonal fits

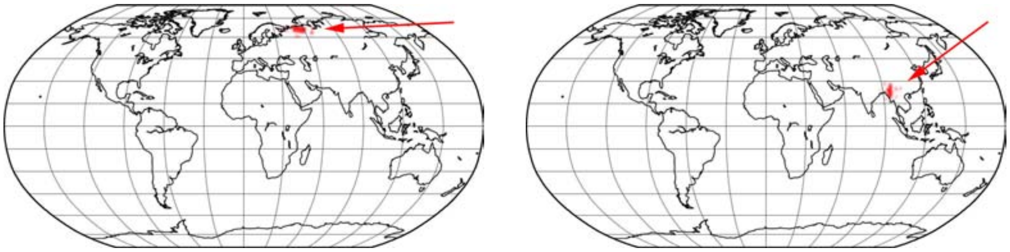
ically improved these measurements and the monthly gravity field estimates can be used to infer the time-variable changes in mass, averaged over arbitrary regions having length scales of a few hundred kilometers and larger. In this section, we will compute the average time-variable signal over a set of small, medium and large sized river basins, using the three gravity fields estimates, and compare the results with the WaterGAP model [13]. WaterGAP stands for "Water - Global Assessment and Prognosis" and was developed with an aim to provide a basis to compare and assess current water resources and water use in different parts of the world, and to provide an integrated long-term perspective of the impacts of global change on the water sector. Swenson et. al. [45] developed techniques for creating spatial averaging kernels which isolate the gravity signal of individual regions while simultaneously minimizing the contamination from surrounding glacial, hydrological and oceanic gravity signals. Using the techniques described in [45], we compute the signal average over the following basins for RL04, REG_{LR} and REG_{S} time series.

- Small basins

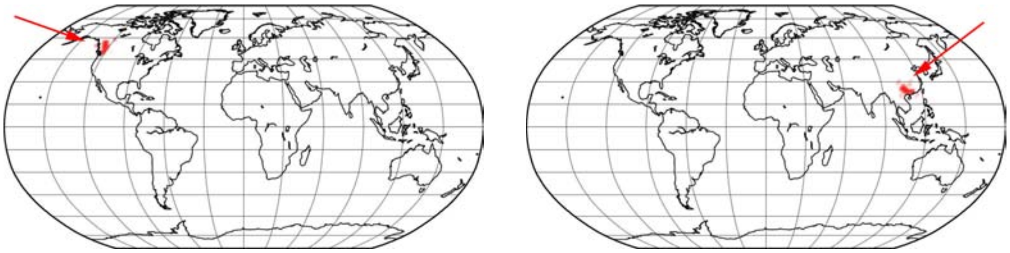
  The small basins chosen for the analysis were Pechora, Irrawaddy, Fraser and Pearl River basins. Figure (4.17) shows the locations and the sizes of these basins.

- Meduim sized basins

  The medium size basins chosen for the analysis were Mekong, Salween, Columbia and Ganges river basins. Figure (4.18) shows the locations



(a) Pechora (b) Irrawaddy



(c) Fraser (d) Pearl River

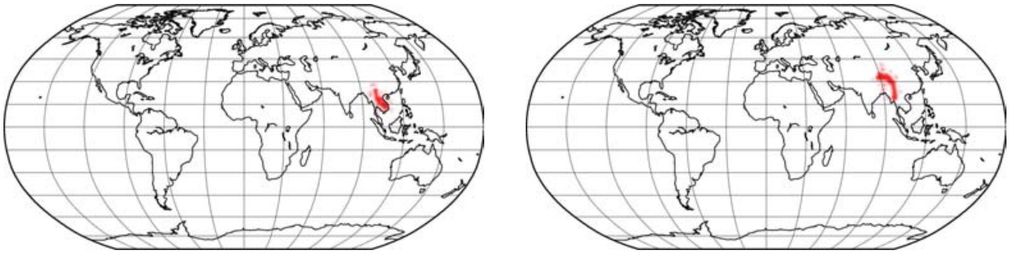
Figure 4.17: Small basins

and the sizes of these medium size basins.

- Large basins

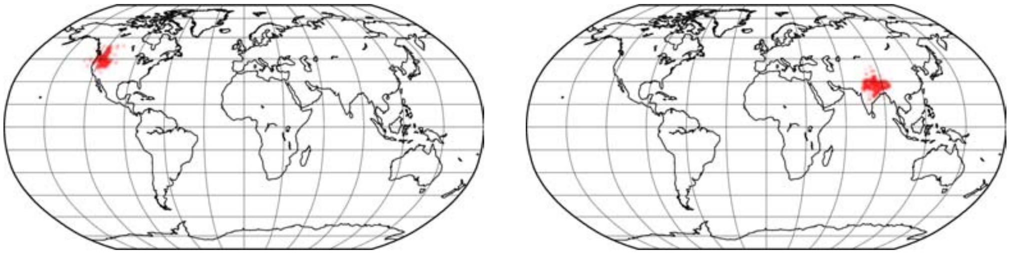
  The large size basins chosen for the analysis were Mississippi, Parana, Amazon and Niger basins. Figure (4.19) shows the locations and the sizes of these basins.

Figures (4.20), (4.21), and (4.22) are the time-series plots for the average signal over the small, medium and large basins respectively. A Gaussian smoothing of 300 km is applied to the RL04 gravity estimates and no smoothing is applied to the regularized time series. It can be observed from the figures that both the regularized series agree with the WaterGAP model for the small



(a) Mekong

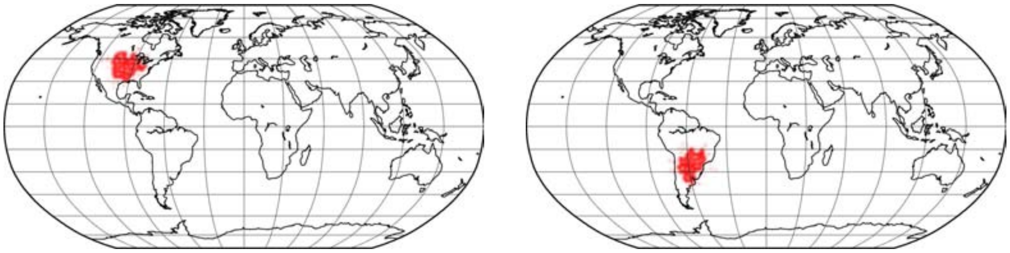
(b) Salween



(c) Columbia

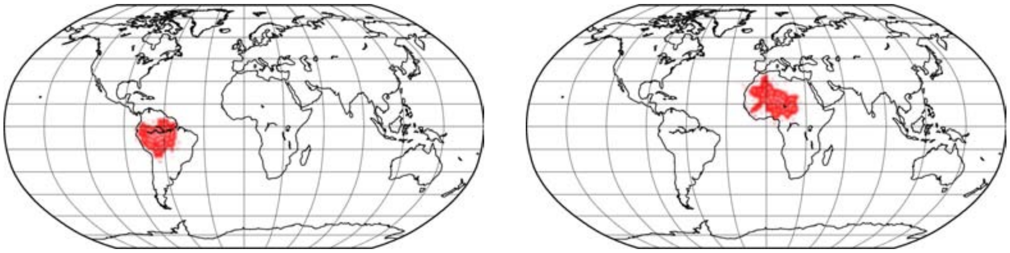
(d) Ganges

Figure 4.18: Medium sized basins



(a) Mississippi

(b) Parana



(c) Amazon

(d) Niger

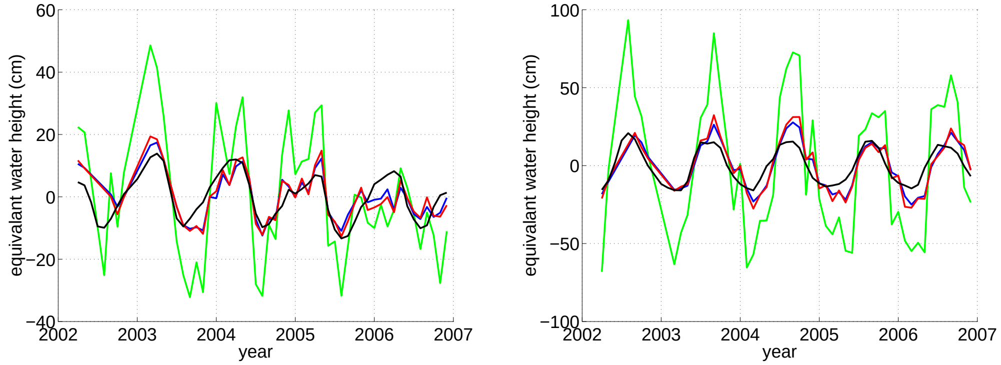
Figure 4.19: Large basins

and the medium basins. The average signal over the large basins in RL04 time series agree very well with the regularized time series. This suggests that while it is very difficult to extract the signal over the small and medium basins using the RL04 estimates because of the presence of errors, the regularized fields is successful in extracting that signal. The agreement of average signal of the regularized solutions with the RL04 solutions suggests that the is no loss of signal in the large basins. There is not appreciable difference between the signals in REG_{LR} and REG_S time series.

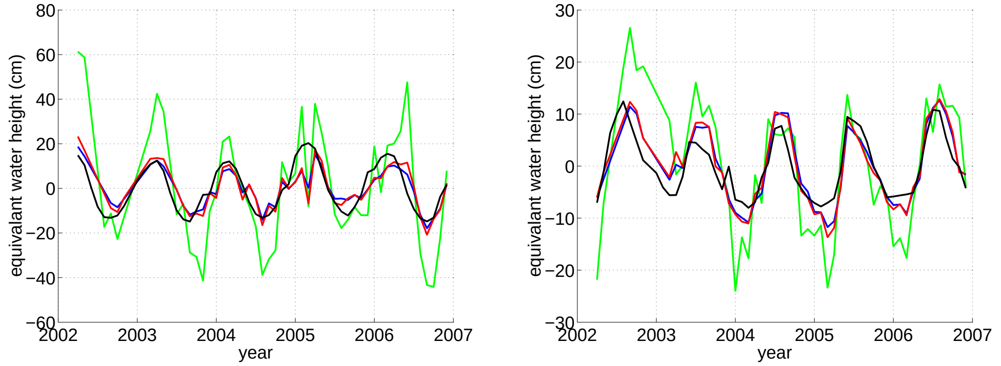
4.7 The Great Sumatra Earthquake signal

The Sumatra-Andaman earthquake of 2004 generated tsunami waves that claimed hundreds of thousands of lives, and permanently changed geography of the Sumatra-Andaman region, raising islands by up to 20 meters [8]. The ruptures extended over approximately 1800 km in the Andaman and Sunda subduction zones. Because of the magnitude of this seismic event and due to solid Earth deformation associated with it, the GRACE products is able to reveal both coseismic and post seismic changes in the gravity field due to this major event.

Figure (4.23) shows the map of mass flux of the difference between the arithmetic mean of the gravity solutions in 2003-2004 and that of the solutions in 2005-2006 for the two time series compared to the co-seismic model of the earthquake [8]. It is expected that the difference between the two year means before and after the earthquakes should capture the co-seismic



(a) Pechora (b) Irrawaddy



(c) Fraser (d) Pearl River

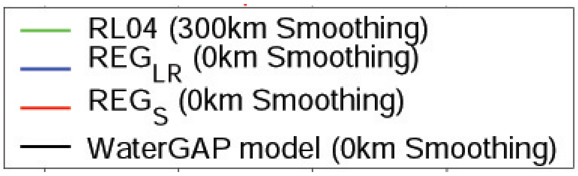
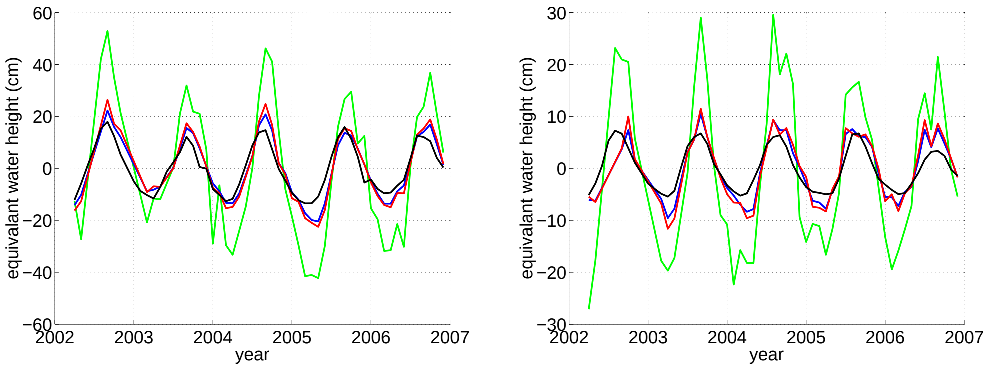
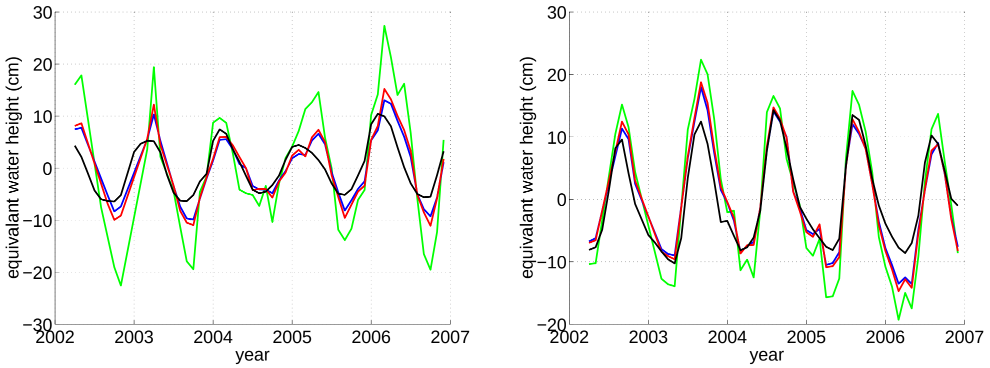


Figure 4.20: Basin signal average comparison over small basins. Red line represents the REG$_S$ solutions, the black line represents the WaterGAP model, the green line represents the REG$_{LR}$ solutions and the blue dotted line represents the RL04 solutions. The RL04 solution is smoothed using a 300km Gaussian filter and no smoothing is applied to the other solutions



(a) Mekong (b) Salween



(c) Columbia (d) Ganges

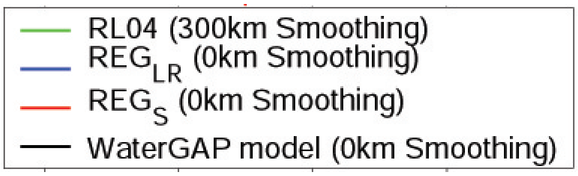
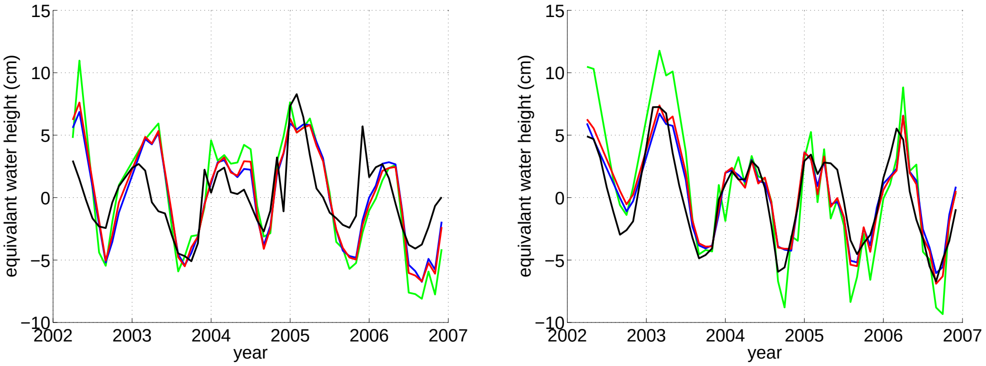
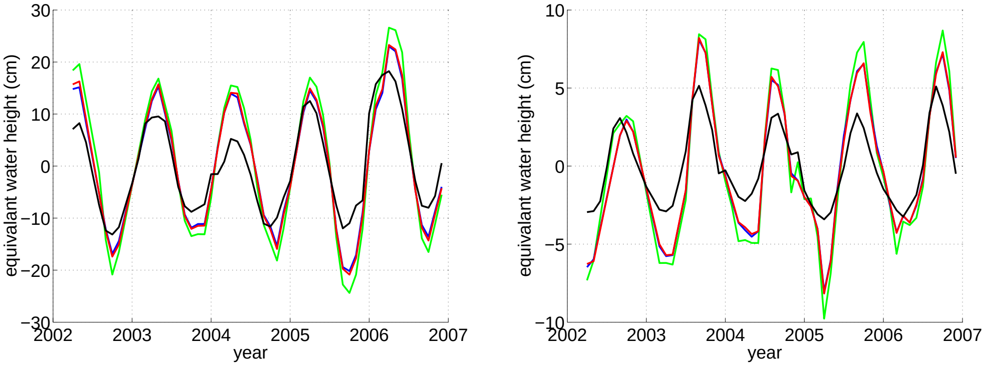


Figure 4.21: Basin signal average comparison over medium sized basins. Red line represents the REG$_S$ solutions, the black line represents the WaterGAP model, the green line represents the REG$_{LR}$ solutions and the blue dotted line represents the RL04 solutions. The RL04 solution is smoothed using a 300km Gaussian filter and no smoothing is applied to the other solutions



(a) Mississippi (b) Parana



(c) Amazon (d) Niger

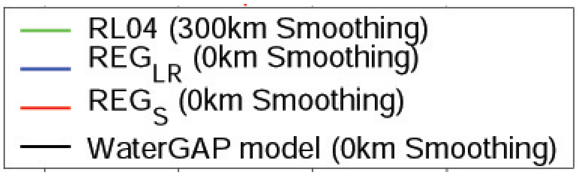
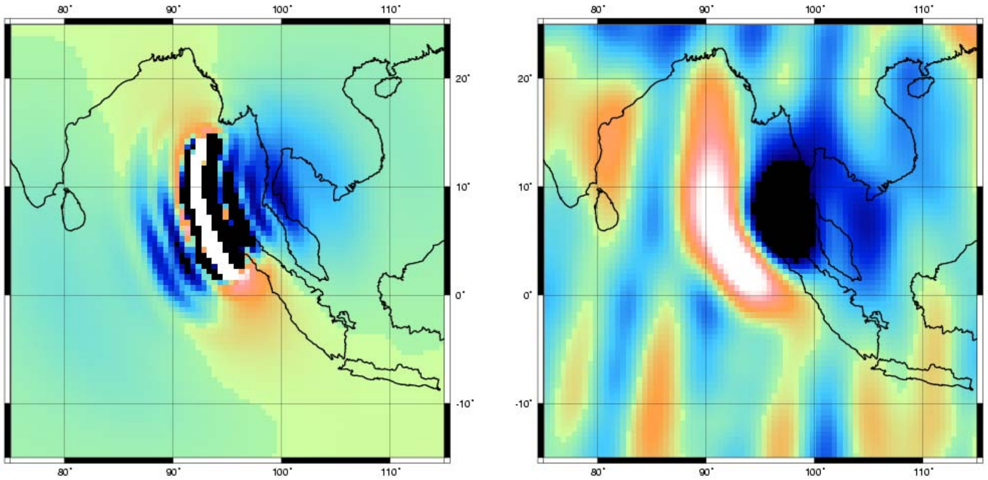
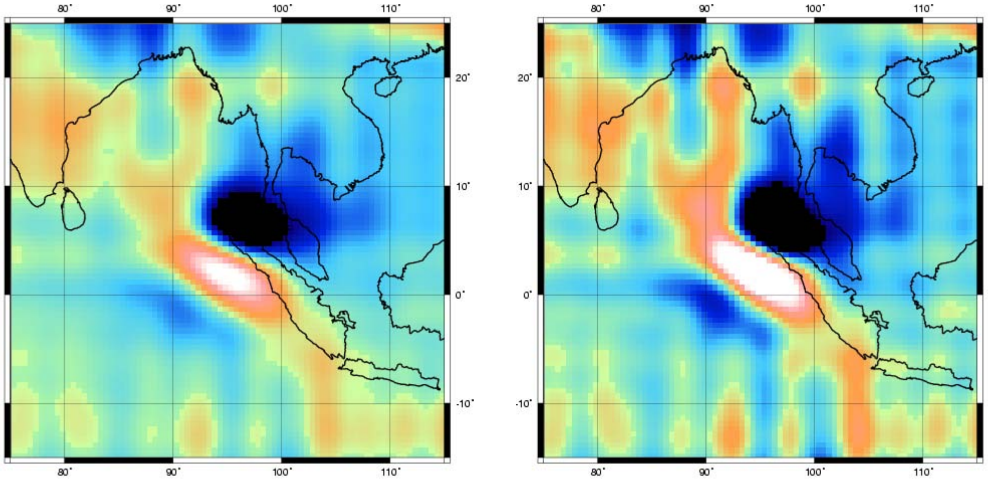


Figure 4.22: Basin signal average comparison over large basins. Red line represents the REG$_S$ solutions, the black line represents the WaterGAP model, the green line represents the REG$_{LR}$ solutions and the blue dotted line represents the RL04 solutions. The RL04 solution is smoothed using a 300km Gaussian filter and no smoothing is applied to the other solutions



(a) Co-seismic Model [15]

(b) RL04 - (350 km smoothing)



(c) REG$_{LR}$ - (No smoothing)

(d) REG$_{S}$ - (No smoothing)

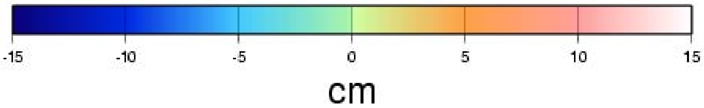


Figure 4.23: (a) shows the model of the co-seismic Great Sumatra Earthquake [15]. (b), (c) and (d) show the difference between respective the averages of (2003-2004) and (2005-2006)

signal of the earthquake. Figure (4.23b) is the difference map for RL04 at 350km Gaussian smoothing and figures (4.23c) and (4.23d) are the difference map for the regularized time series with no smoothing applied. As expected from the post-fit analysis the earthquake signal is not completely captured by the regularized solutions. Though, REG_S captures more earthquake signal as compared to REG_{LR} solutions, there is clearly a loss in this signal in both the solutions. The Sumatra earthquake signal has a large magnitude with a very small spatial extent and requires very high degree and order harmonics to describe the signal. This regularization technique, which is designed for time variable gravity, thus fails to capture the entire earthquake signal.

Chapter 5

Summary

The gravity model resulting from the GRACE estimation problem is susceptible to geographic errors that show up as north-south stripes in the global map of the monthly geoid variability. This study used regularization techniques to remove these systematic errors in the GRACE gravity solutions while ensuring that there was no signal attenuation.

5.1 Summary of contributions

- The most commonly used regularization matrix for gravity related problems is a one that is based on the "Kaula's rule". But as discussed in the chapter 3, such a matrix would potentially attenuate the time-variable signal. The regularization matrix M_D, used in this study, was empirically designed for the time-variable gravity estimation problem such that there is no attenuation of signal and that the solutions fit the observations as well as the unconstrained solutions. The design decisions for M_D were primarily based on the observation that the systematic errors in the geo-potential coefficients from GRACE start above order 8 [46]. The design of the regularization matrix depends on both the observations and on the observation noise in the RL04 estimates. Any change in the background model or pre-processing of the data will change the error characteristics of the solution. Hence, a redesign of the regularization matrix will be required with any changes in the background models or pre-processing of data.

- The algorithm for the regularized solution of the time variable gravity estimation problem was developed as a combination of orthogonal transformation and computationally inexpensive variant of Tikhonov regularization. The orthogonal transformation approach is different from the previous studies of regularization of gravity field estimation problems which used the normal equations approach. The regularization in this study was implemented in the current GRACE processing framework by showing the equivalence of the orthogonal transformation approach to that from the normal equations.

- A projection method implementation of L-curve called L-ribbon was used as the parameter choice method to find the optimal value of the regularization parameter, μ. The Morozov's Discrepancy Principle was not used for parameter choice because it requires knowledge of the norm of the true observation error, which is unknown for the GRACE problem. The Quasi-Optimality Criterion method was not used because of the difficulty in finding the optimal regularization parameter due to the presence of multiple local minimums in the parameter search curve as discussed in section 2.5. The projection method implementation of the Generalized Cross-Validation (GCV) for the GRACE estimation problem had convergence issues making the problem computationally more expensive than the L-ribbon implementation and producing solutions with more reduced signal content as compared to those obtained using the L-ribbon implementation as discussed in the section 3.5.

The use of L-ribbon method requires the computation of a bidiagonal matrix using Lanczos bidiagonalization. The computation of this Lanczos bidiagonalization was made feasible for GRACE by using the parallel computing resources available at the Texas Advanced Computing Center (TACC) at the University of Texas at Austin. The Lanczos bidiagonalization computation was parallelized and implemented with re-orthogonalization on a massively parallel Linux cluster named *lonestar* at TACC. All the other programs associated with the regularization, like computation of the L-ribbon, were implemented on a parallel computing architecture.

- This study uses the post-fit residuals in the range acceleration domain as a unique way to discuss the signal and noise content in the regularized gravity estimates.

5.2 Summary of results

- Error Reduction

  The variability maps of equivalent water heights obtained from the regularized solutions show that the "stripes" are significantly reduced in the monthly regularized solutions as compared to the unconstrained RL04 estimates. The regularized solutions obtained are of uniform quality over the entire mission.

  There is a high correlation of about 0.8 up to degree 14 between the regularized solutions and the RL04 solutions which suggests that the regularized solution captures all the time-variable signal in the low degrees.

  The statistics of the residuals with respect to seasonal fits over the oceans show significant noise reduction over the oceans as compared to the unconstrained RL04 solutions.

- Signal Content

  The basin average analysis for the small and medium sized basins show good agreement with the WaterGAP hydrological model suggesting that the regularized solutions capture the hydrology signal in the small and medium basins without the need for post-processing. The regularized solutions show a good agreement with the RL04 solutions over the large basins, suggesting that there is little or no loss of signal. The literature warns of deficiencies in the hydrological models and that GRACE gravity models captures more signal amplitude than that predicted by the models. But this comparison is very difficult in the small basins with the RL04 gravity field estimates because of the presence of large error stripes at those small spatial scales. The regularized GRACE gravity estimates make these comparisons feasible in the small basins without post-processing.

The residuals over the oceans with respect to the seasonal sinusoids are highly reduced, in comparison with the unconstrained RL04 solutions, to an extent where the RMS residual over the oceans reveals an expected structure. This RMS residual, which contains the errors and the signals not captured by the sinusoids, is low at the equator and high at the high latitudes. The signal in the southern sea, south west of Australia, shows up in the regularized solutions but is masked by the errors in the unconstrained RL04 solutions, which require additional post-processing to reveal these signals.

The co-seismic signal of the great Sumatra earthquake is left over in the post-fit residuals of the regularized solutions and regularization fails to capture that signal. This observation is also supported by the maps of the difference between the two year means, before and after the earthquake. The Sumatra earthquake signal has a large magnitude with a very small spatial extent and requires very high degree and order harmonics to describe the signal. This regularization technique, which is designed for time variable gravity, thus fails to capture the entire earthquake signal.

- Post-fit residuals

  The post-fit analysis confirms that no time-variable signal is left in the residuals, computed after fitting the solution to the data. The post-fit analysis suggests that regularization is superior to Gaussian smoothing as a method for removing the errors in the results from the monthly solutions. The map of residuals shows geo-spatial correlation, indicating that there is signal loss associated with Gaussian smoothing.

5.3 Recommendations for GRACE

- Any improvement in the background models or any preprocessing of the data will change the error characteristics of the solution. Since, the design of the regularization matrix depends on these error characteristics, a redesign of the regularization matrix will be required.
- Since, regularization balances the size of the observation residual ($\|H\hat{x}-$

$y\|$) and the size of the solution ($\|\hat{x}\|$), there is a potential for signal attenuation in the areas of high variability signal. Thus, a hydrological background model should be applied to reduce the size of the solution ($\hat{x}$) to be estimated thereby avoiding any possible signal attenuation in the areas of high variablity signal.

- The regularization matrix used in this study is designed for time variable gravity and thus constrains the high degree and order coefficients. There is a possibility for signal attenuation in the areas of small spatial scales during regularization. Thus, in future studies, the events like an earthquake need to be modeled as a step function in the background model thus removing the large signal with a small spatial extent from the time variable signal that GRACE estimates. The regularized solutions will then capture any error in the co-seismic earthquake model and the post-seismic relaxation signal which is of a much lesser magnitude.
- This regularization works very effectively for the time variability signal in the GRACE solutions and captures the signal with minimal loss while introducing minimal errors. The REG$_S$ class of regularized solutions, though not ready for the replacement of RL04 solutions, can be used for scientific analysis without the need for any post processing.

Appendices

Appendix A

Estimation theory

Early 17th century marked the beginning of orbit prediction with accurate and comprehensive astronomical and planetary observations of Tycho Brahe and with Johannes Kepler's laws of planetary motion that could match Brahe's data to the true elliptical shape of the Mars orbit. Issac Newton established the mathematical foundation from which Kepler's Laws could be derived. The invention of the process of least squares analysis by Carl Friedrich Gauss, later in that century provided a computational basis for orbit prediction. The goal of orbit prediction is to obtain accurate ephemeris (position and velocities) of an orbiting satellite using a sequence of observations. A predicted observation is produced by the integration of the equations of motion of a satellite from a reference epoch to the time of true observation using a model to approximate how the satellite was observed. The difference between the predicted observations and the true observations, called residuals, are minimized in an estimation process that determines the kinematic and dynamic parameters that describe the satellite's ephemeris and those which describe the participating models.

A.1 The geopotential model

Earth's geopotential is represented through the use of spherical harmonics by the expression, [29]

$$U(r,\phi,\lambda,t) = \frac{\mu}{r}\sum_{l=2}^{\infty}\sum_{m=0}^{l}\left(\frac{a_e}{r}\right)^l \bar{P}_{lm}(\sin\phi)[\bar{C}_{lm}(t)\cos m\lambda + \bar{S}_{lm}(t)\sin m\lambda] \tag{A.1}$$

where λ, ϕ and r represent the spherical coordinates at which the potential is evaluated. l and m represent the degree and order of the spherical harmonic expansion, and $\bar{P}_{lm}$ are the normalized Legendre polynomial functions. The constants μ and a_e represent the geocentric gravitational constant and equatorial radius of the Earth. The normalized spherical harmonic coefficients $\bar{C}_{lm}$ and $\bar{S}_{lm}$ can represent the mean and time variable components of the potential. This expression assumes an Earth-fixed reference frame. As illustrated in the figure A.1, the gravity coefficients are typically characterized into three classes: zonals, tesserals and sectorials.

The value of these coefficients are determined by measuring the path of an orbiting satellite and comparing the measurements against its computed or predicted path as determined by a reference gravity model. In the case of GRACE, it depends on the GPS observations and the KBR tracking data, to determine the position and velocity of each satellite in space. Since the positions and velocities are not measured directly, a second model called the measurement model needs to be created in order to relate these observations to

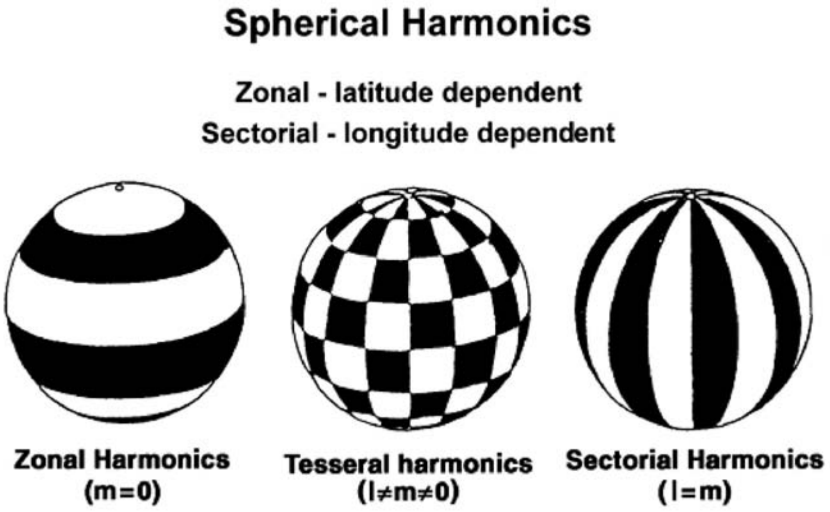


Figure A.1: Illustration of spherical harmonics [22]

the dynamical model given by the equation (A.1). The correct models ensure that the computed orbit is very close to the observed orbit.

A.2 Least squares estimation

The equations of motion of a satellite in orbit are represented as follows in a vector form as a system of linear first order equations, with time t as an independent variable:

$$\dot{X}(t) = F[X(t), t] \quad \text{and} \quad \dot{X}^*(t) = F[X^*(t), t] \tag{A.2}$$

for true and nominal state respectively, where:

$$X = \begin{bmatrix} \vec{r} \\ \vec{v} \\ \vec{\alpha} \end{bmatrix} \quad \text{(n} \times \text{1 vector)}$$

$\vec{r}$ = satellite position (3 × 1 vector)

$\vec{v}$ = satellite velocity (3 × 1 vector)

$\vec{\alpha}$ = vector of model parameters

F = derivatives of the state vector X (n × 1 vector)

with initial conditions $X(t_0)$ and $X^*(t_0)$

The nominal state $X^*(t)$, refers to the state of the satellite that is the previously known best estimate. The true state, $X(t)$ is a combination of the nominal and some deviation, $x(t)$

$$X(t) = X^*(t) + x(t) \tag{A.3}$$

where $x(t)$ is a $(n \times 1)$ vector of deviations away from the computed nominal state $X^*(t)$. We can rearrange equation (A.3) and using equation (A.2) and taking the first derivative with respect to time, we get

$$\begin{aligned} \dot{x}(t) &= \dot{X}(t) - \dot{X}^* t \\ &= F[X(t), t] - F[X^*(t), t] \\ &= F[X^*(t) + x(t), t] - F[X^*(t), t] \end{aligned}$$

Expanding this equation in the Taylor series expansion about the nominal trajectory and ignoring higher order terms yields,

$$\begin{aligned}\dot{x}(t) &= F[X^*(t),t] + \left.\frac{\partial F[X^*(t),t]}{\partial X^*(t)}\right|_* x(t) + \text{H.O.T} - F[X^*(t),t] \\ &= A(t)x(t)\end{aligned}$$

where

$$A(t) = \frac{\partial F[X^*(t),t]}{\partial X^*(t)} = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \frac{\partial F_1}{\partial X_2} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \frac{\partial F_2}{\partial X_1} & \frac{\partial F_2}{\partial X_2} & \cdots & \frac{\partial F_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial X_1} & \frac{\partial F_n}{\partial X_2} & \cdots & \frac{\partial F_n}{\partial X_n} \end{bmatrix}_* \quad n \times n \text{ matrix} \tag{A.4}$$

The linearized differential equation (A.4) has a solution,

$$x(t) = \Phi(t,t_0)x(t_0) \tag{A.5}$$

where $x(t_0)$ is the value of $x(t)$ at the epoch t_0 and $\Phi(t,t_0)$ is the state transition matrix which relates a deviation to the state, $x(t)$, at some time t to the state at t_0 [18]. The matrix $\Phi(t,t_0)$ satisfies the differential equation

$$\dot{\Phi}(t,t_0) = A(t)\Phi(t,t_0) \tag{A.6}$$

with the initial condition of $\Phi(t_{0,0}\,) = I$, the identity matrix.

If there are p observations taken at time t, they can be represented by a (p x 1) vector $Y(t)$. We then define an observation-state equation as,

$$Y(t) = G[X(t),t] + \epsilon(t) \tag{A.7}$$

where, $G[Xt,t]$ is a $(p \times 1)$ vector that represents the mathematical model of the satellite observations and the $(p \times 1)$ vector ϵ represents the errors in the observation models. Using equation A.3 and expanding the equation A.7 in a Taylor series expansion and dropping the terms higher than first order, we can obtain the relation between the observation residual $y(t)$ and deviation $x(t)$.

$$\begin{aligned} Y(t) &= G[X^*(t)+x(t),t]+\epsilon(t) \\ Y(t) &= G[X^*(t)]+\left.\frac{\partial G[X^*(t),t]}{\partial X^*(t)}\right|_* x(t)+\epsilon(t) \\ Y(t)-G[X^*(t)] &= \left.\frac{\partial G[X^*(t),t]}{\partial X^*(t)}\right|_* x(t)+\epsilon(t) \\ y(t) &= \tilde{H}(t)x(t)+\epsilon(t) \qquad \text{(A.8)} \end{aligned}$$

where $\tilde{H}(t)$ is defined by,

$$\tilde{H}(t) = \left.\frac{\partial G[X^*(t),t]}{\partial X^*(t)}\right|_* x(t) \qquad \text{(A.9)}$$

and $\epsilon(t)$ contains the errors due to linearization and the observation errors. From equation (A.5) and equation (A.9) we get,

$$y(t_i) = \tilde{H}(t_i)\Phi(t_i,t_0)x(t_0)+\epsilon(ti) \qquad \text{(A.10)}$$

If we let $H(t_i) = \tilde{H}(t_i)\Phi(t_i,t_0)$, we then get an expression that maps an observation taken at time t_i to the initial time t_0.

$$y(t_i) = \tilde{H}(t_i)\Phi(t_i,t_0)x(t_0)+\epsilon(ti) \qquad \text{(A.11)}$$

where, $y(t_i)$ and $\epsilon(t_i)$ are $(p\times 1)$ vectors, $x(t_0)$ is (n x 1) and $H(t_i)$ is a $(p\times n)$ matrix. A set of observations taken at times $[t_1,t_2,\cdots t_k]$ can all be represented by one equation in the following matrix form.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} H(t_1) \\ H(t_2) \\ \vdots \\ H(t_k) \end{bmatrix} x_0 + \begin{bmatrix} \epsilon(t_1) \\ \epsilon(t_2) \\ \vdots \\ \epsilon(t_k) \end{bmatrix} \tag{A.12}$$

which is compactly written as,

$$y = Hx_0 + \epsilon \tag{A.13}$$

The correction $\hat{x}_0$ is estimated and is added to the nominal state vector $X^*(t)$ at the initial epoch at t_0. This correction is obtained by minimizing the weighted sum of the square of the observation residuals as defined by the following performance index [51].

$$\begin{aligned} J(x) &= \sum \epsilon(t_i) w_i \epsilon(t_i) \\ &= \epsilon^T W \epsilon \end{aligned} \tag{A.14}$$

where W is a diagonal matrix containing assigned observation weights w_i applied to each observation i. We can substitute the equation (A.13) in the equaion (A.14) and obtain the following expression:

$$J(x) = (y - Hx)^T W (y - Hx) \tag{A.15}$$

Setting the first variation of equation (A.15) equal to zero, results in the following normal equations of the linear system.

$$(H^T W H)\hat{x} = H^T W y \tag{A.16}$$

If the matrix (H^TWH) is positive definite, it will have an inverse and the solution is written as

$$\hat{x} = (H^TWH)^{-1}H^TWy \tag{A.17}$$

The variance and covariance of the unbiased estimated correction vector $\hat{x}$ is given by [51]

$$\begin{aligned} P &= E\left[(\hat{x} - E[\hat{x}])\right] E\left[(\hat{x} - E[\hat{x}])^T\right] \\ &= E\left[(\hat{x} - x)\right] E\left[(\hat{x} - x)^T\right] \\ &= (H^TWH)^{-1}H^TWE[\epsilon\epsilon^T]WH(H^TWH)^{-1} \end{aligned} \tag{A.18}$$

If the weighting matrix W is chosen such such that it is the inverse of the observation covariance, i.e. $W = \{E[\epsilon\epsilon^T]\}^{-1}$, then the equation (A.18) reduces to

$$P = (H^TWH)^{-1} \tag{A.19}$$

A.2.1 Apriori estimate

If $\bar{x}$ is an *apriori estimate* and its corresponding covariance is $\bar{P}$ called *apriori* covariance, then

$$\begin{aligned} y &= Hx + \epsilon \quad : W \\ \bar{x} &= x + \bar{\epsilon} \quad : \bar{W} = \bar{P}^{-1} \\ \begin{bmatrix} y \\ \bar{x} \end{bmatrix} &= \begin{bmatrix} H \\ I \end{bmatrix} x + \begin{bmatrix} \epsilon \\ \bar{\epsilon} \end{bmatrix} \\ y' &= H'x' + \epsilon' \quad : W' \end{aligned} \tag{A.20}$$

where $y' = \begin{bmatrix} y \\ \bar{x} \end{bmatrix}$, $\epsilon' = \begin{bmatrix} \epsilon \\ \bar{\epsilon} \end{bmatrix}$, $H' = \begin{bmatrix} H \\ I \end{bmatrix}$ and $W' = \begin{bmatrix} W & 0 \\ 0 & \bar{P}^{-1} \end{bmatrix}$. If we redefine the performance index as,

$$\begin{aligned} J' &= \epsilon'^T W' \epsilon' \\ &= \begin{bmatrix} \epsilon^T & \bar{\epsilon}^T \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & \bar{P}^{-1} \end{bmatrix} \begin{bmatrix} \epsilon \\ \bar{\epsilon} \end{bmatrix} \\ &= = \epsilon^T W \epsilon + \bar{\epsilon}^T \bar{P}^{-1} \bar{\epsilon} \\ &= = (y - Hx)^T W (y - Hx) + (\bar{x} - x)^T \bar{P}^{-1} (\bar{x} - x) \qquad \text{(A.21)} \end{aligned}$$

Setting the first variation equation to zero gives the adjusted weighted least squares estimated for the inclusion of the *a priori estimate* as

$$\hat{x} = \left[(H^T W H) + \bar{P}^{-1}\right]^{-1} \left[H^T W y + \bar{P}\bar{x}\right] \qquad \text{(A.22)}$$

and with $W = \{E[\epsilon\epsilon^T]\}^{-1}$, the resultant covariance is,

$$P = \left[(H^T W H) + \bar{P}^{-1}\right]^{-1} \qquad \text{(A.23)}$$

A.3 Solution via orthogonal transformation

The method of normal equations is one technique commonly used to solve least squares systems. An alternative to this approach involves the introduction of the orthogonal transformation, also known as QR factorization. This alternate approach is used to avoid some of the numerical problems encountered in the normal equations approach. The method obtains solution by applying successive orthogonal transformations to the information array,

(H, y). Consider the performance index given in equation (A.15), which obtains the solution to the least squares problem (as well as the minimum variance and the maximum likelihood estimate [51] which can be rewritten with $-\epsilon$ as

$$J(x) = (Hx - y)^T W (Hx - y) \tag{A.24}$$

In this approach, in order to obtain the solution $\hat{x}$ that minimizes the equation (A.24), we introduce the $m \times m$ orthogonal matrix Q. This orthogonal matrix has the following properties:

1. $QQ^T = I$
2. $Q^{-1} = Q^T$ hence, $Q^T Q = I$
3. If Q_1 and Q_2 are orthogonal matrices, the so is $Q_1 Q_2$
4. For any vector x: $\|Qx\| = \|x\| = (x^T x)^{\frac{1}{2}}$

   Multiplying by Q does not change the Euclidean norm of the vector.
5. If ϵ is a m vector of random variables with $\epsilon \sim (0, I)$ (i.e. $E(\epsilon) = 0$ and $E(\epsilon\epsilon^T) = I$), then $\bar{\epsilon} = Q\epsilon$ has the same properties,

   $E(\bar{\epsilon}) = QE(\epsilon) = 0$ and $E(\bar{\epsilon}\bar{\epsilon}^T) = QE(\epsilon\epsilon^T)Q^T = I$

Thus, equation (A.24) can be expressed as

$$\begin{aligned} J(x) &= \left\| QW^{\frac{1}{2}}(Hx - y) \right\|^2 \\ &= (Hx - y)^T W^{\frac{1}{2}} Q^T Q W^{\frac{1}{2}} (Hx - y) \end{aligned} \tag{A.25}$$

Select Q such that,

$$\begin{aligned} QW^{\frac{1}{2}}H &= \begin{bmatrix} R \\ O \end{bmatrix} \\ \text{and define } \; QW^{\frac{1}{2}}y &= \begin{bmatrix} \mathbf{b} \\ e \end{bmatrix} \end{aligned} \tag{A.26}$$

where, R is a $n \times n$ upper-triangular matrix of rank n

O is a $(m-n) \times n$ null matrix

$\mathbf{b}$ is a $n \times 1$ column vector

e is a $(m-n) \times 1$ column vector

If $m > n$ and H is of rank n, then using equations (A.26) we can write equation (A.24) as

$$J(x) = \left\| \begin{bmatrix} R \\ O \end{bmatrix} x - \begin{bmatrix} \mathbf{b} \\ e \end{bmatrix} \right\|^2 \tag{A.27}$$

Which leads to

$$J(x) = \|Rx - \mathbf{b}\|^2 + \|e\|^2 \tag{A.28}$$

Since only the first term is a function of x, the value of x that minimizes $J(x)$ is obtained by requiring that

$$R\hat{x} = \mathbf{b} \tag{A.29}$$

and the minimum value of the performance index becomes

$$J(\hat{x}) = \|e\|^2 = \left\| W^{\frac{1}{2}}(H\hat{x} - y) \right\| \tag{A.30}$$

A.4 Optimal weighting

The process of estimation of the gravity field incorporates a large number of data points collected by different instruments. The accuracy and the precision of these instruments are not always the same. For example, in case of the GRACE mission, the K-band ranging instrument is extremely precise and records the range measurements to an accuracy of less than 10 micrometers, while the GPS measurements are at sub-centimeter level precision. Both the measurements are important in the process of gravity field estimation by GRACE, but KBR measurement is the primary observable for the GRACE mission because of its sensitivity to the Earth's gravity field. It is therefore important that the less sensitive and the less precise GPS measurement should not be given the same weight as the KBR measurement during the estimation process. Even the data within the same data set at different times might be of varying accuracy because of instrument errors and temporary malfunctions. When combining these data sets of mixed quality, we should give more weight to the observations with higher quality than those with lower quality.

Born et. al. [3] and Tapley et. al. [48] have developed an approach that essentially predicts the post-fit residuals of each dataset and assigns a weight based on how well the computed estimates fit the data. Starting with predefined weights and using equation (A.26) and equation (A.29) we find the estimate $\hat{x}$ which gives us the error estimate using equation (A.30). This estimate of error, $J(\hat{x})$ is used to calculate the new weight for the given data set as,

$$f = \frac{m}{J(\hat{x})} \tag{A.31}$$

where m is the number of observations in the given data set.

A.5 GRACE processing scheme

The complex process of collection and processing of the GRACE data involves a number of agencies around the world. The two broad categories that the tasks are divided into are Mission Operations System (MOS) and Science Data Systems (SDS). The majority of MOS duties are conducted by the German Space Operations Center (GSOC) which involve the maintenance of the satellites and collection of the raw science data. One component of SDS, that involves initial processing of the raw science data including the basic cleanup of the data and packaging in the predefined formats, is done by the Jet Propulsion Laboratory (JPL). The other component of SDS that involves the interpretation and combination of these data files into gravity field models, which is the primary science product of the GRACE mission, is performed mostly at the Center of Space Research (CSR) at The University of Texas at Austin and at Geo Forschungs Zentrum (GFZ).

The GRACE processing scheme at UT/CSR begins with the creation of a series of GPS double-differenced observations that are sampled at 30 sec intervals and the K-band range-rate observations that are sampled at 5 sec intervals. Using these observations, a set of measurement partials are created for each data type based on the process outlined in section A.2. The integration interval for the batch process is mostly used as 1 day unless the day has to be broken into multiple arcs due to some technical problems like loss of data or poor data quality. This process makes use of the Multi-Satellite Orbit Determination Program (MSODP) [42] and involves the numerical integration of hundreds of thousands of partial differential equations. These partial differential equations are then accumulated using Advanced Equation Solver for Parallel Systems (AESoP) [22] which employs the orthogonal transformations to convert the large $m \times n$ partials matrix to a $n \times n$ upper triangular R matrix, which is used to solve for the gravity field estimate. Typically, one month's worth of GPS and KBR data is combined into a single run to estimate a monthly gravity field model. These monthly accumulation (R) files can be combined to create an annual or long term mean gravity models.

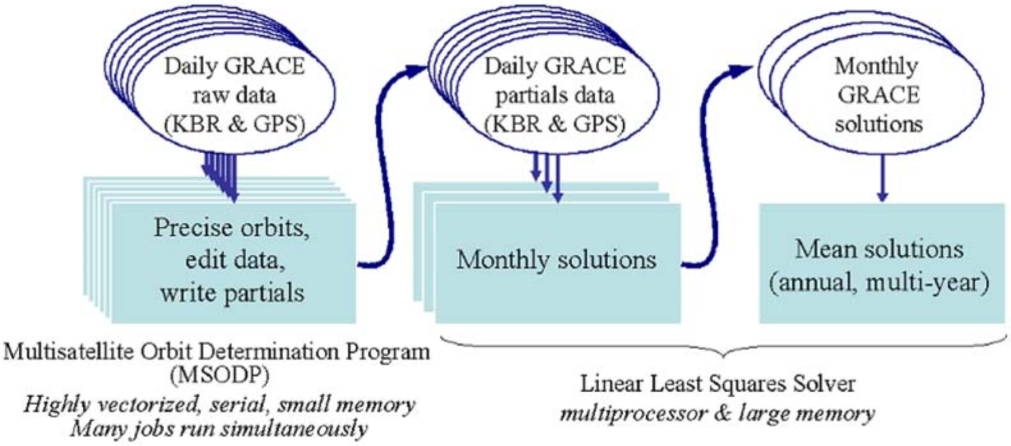


Figure A.2: GRACE data processing scheme [22]

The computational requirements for this estimation process depends on the number of parameters estimated and on the number of observation points, which is a function of the sampling interval. A gravity filed of size 160 × 160 roughly has about 26,000 (n) parameters. To process to create a single day's measurement partials for a gravity field of this size is an $O(6n)$ operation, requiring a numerical integration of $\sim$ 156,000 differential equations. There are roughly 17200 observations of KBR data and roughly 50,000 observations of GPS data that are used to create the daily partials file for each data set. The total size of these set of daily partials file is $\sim$ 5 gigabytes. If m is the total number of observations ($\sim$ 67000 per day) and n is the total number of parameters to be estimated ($\sim$ 26,000 for a 160 × 160 gravity field), the accumulation process (QR factorization) involves $O(mn^2)$ operations, which translates to over 21 trillion floating point operations (FLOPS). These daily statistics increase to much larger totals when we look at monthly time frames. These numbers do not include the additional storage required for other data files and outputs such as the covariance matrix files and the accumulated R matrix files. Thus, high performance computing and large volume storage are the requirements for GRACE processing.

Appendix B

Parallel computation

The use of parallel processing has become important in the fields of numerical analysis where the problem size is too large for sequential machines. Even though historically, the computation speed of individual processor has been doubling every 18 months, it is the memory requirement of certain algorithms that prevent them from handling large problems. Traditionally, software has been written for serial computation to be run on a single computer having a single processor. An algorithm is broken into a discrete series of instructions which are executed one after another. In a serial algorithm only one instruction may execute at any moment in time. Figure (B.1) illustrates the way a serial algorithm is implemented. On the other hand a parallel computation is the simultaneous use of multiple compute resources to solve a computational problem run using multiple CPUs. A problem is broken into discrete parts that can be solved concurrently and each part is further broken down to a series of instructions. Each of these instructions are executed simultaneously on different CPUs. The compute resources can include a single computer with multiple processors, an arbitrary number of computers connected by a network or a combination of both. A serial algorithm usually limits the performance and the available memory but with parallel computation one is able to solve large problems that usually don't fit on a single processor memory. The parallelization of a computational problem usually demonstrates characteristics such as the ability to be broken apart into discrete pieces of work that can be solved simultaneously, and execute multiple program instructions at any moment in time. This allows us to solve problems that would otherwise take unreasonable amount of time, thus allowing us to solve more cases of larger problems much faster than a serial algorithm would. Figure (B.2) illustrates the flow of a typical parallel algorithm [37].

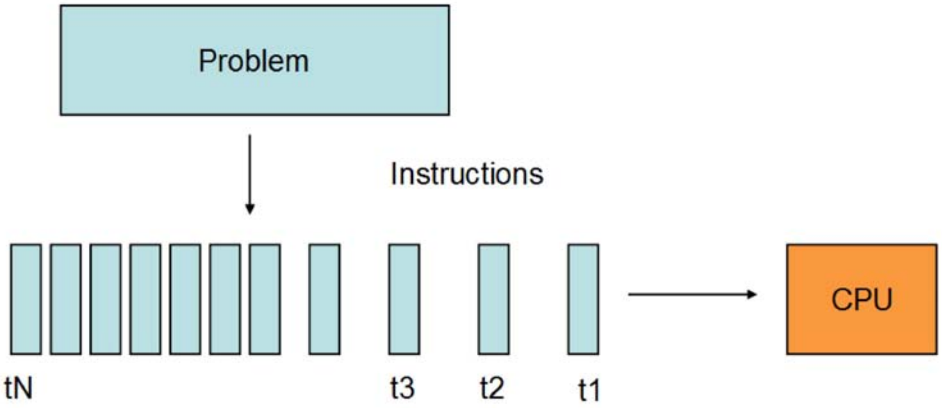


Figure B.1: Illustration of a serial algorithm

B.1 Machine architecture

Parallel computers have evolved into reliable computational resources. Through the years, a variety of systems have been developed to meet the ever increasing demand for more processing power. However, problems of portability in software, both for algorithms and processor communications, across

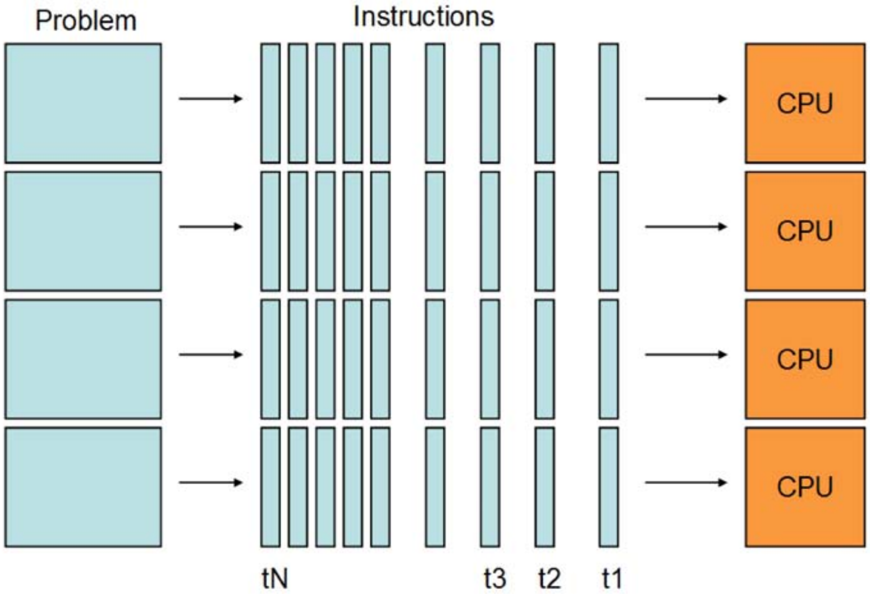


Figure B.2: Illustration of a parallel algorithm architectures, has hindered the ability of parallel computation to compete with conventional serial programming. Until recently, Flynn's taxonomy was commonly used to classify parallel computers into one of the four basic types [17]:

- Single instruction, single data (SISD)

  This is a serial (non-parallel) computer. Single instruction implies that only one instruction stream is being acted on by the CPU during any one clock cycle. Single data implies that only one data stream is being used as input during any one clock cycle. This is the oldest and until recently, the most prevalent form of computer.

- Single Instruction, Multiple Data (SIMD)

  This is a type of parallel computer where single instruction implies that all processing units execute the same instruction at any given clock cycle. Multiple data refers to the fact that each processing unit can operate on a different data element This type of machine typically has an instruction dispatcher, a very high-bandwidth internal network, and a very large array of very small-capacity instruction units. This type of computer is best suited for specialized problems characterized by a high degree of regularity,such as image processing.

- Multiple Instruction, Single Data (MISD)

  In this type of computer, a single data stream is fed into multiple processing units. Each processing unit operates on the data independently via independent instruction streams. Few actual examples of this class of parallel computer have ever existed. One such example is the experimental Carnegie-Mellon C.mmp computer (1971).

- Multiple Instruction, Multiple Data (MIMD)

  This type of architecture is the most common in parallel computers these days. Multiple Instruction implies that every processor may be executing a different instruction stream. Multiple Data refers to the fact that every processor may be working with a different data stream.

Since MIMD model is the most popular parallel architecture, the much more relevant way to classify the modern parallel computers is by their memory model, namely, *shared memory* and *distributed memory.* The driving force behind these systems philosophies is that processors must have the ability to communicate with each other in order to cooperatively complete a task.

B.1.1 Shared memory

A shared memory system is the one in which its processors have access to a pool of shared memory. Multiple processors can operate independently but share the same memory resources. The changes in a memory location effected by one processor are visible to all other processors. Figure (B.3) is a schematic diagram of such a system. This configuration allows the processors to communicate with each other through variables stored in a shared address space, however care must be taken that that processors do not access regions

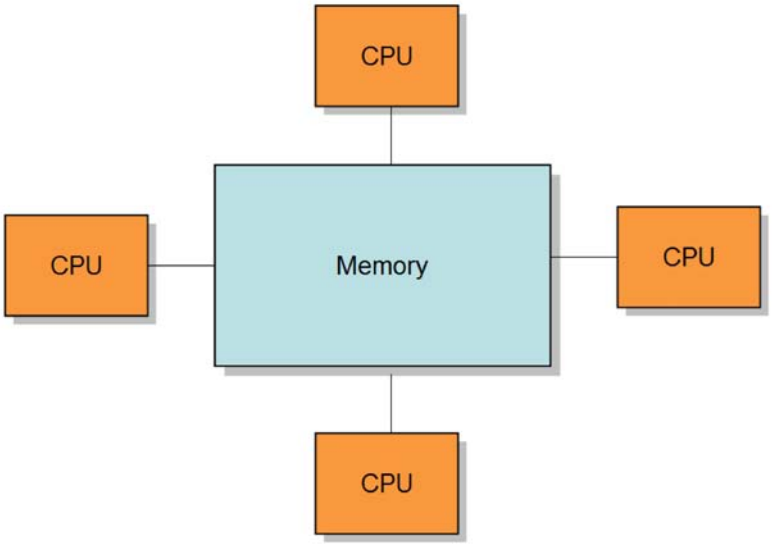


Figure B.3: Illustration of a shared memory architecture

of memory in such a way that errors would occur. This makes referencing data stored in the memory similar to the traditional single processor programs. The advantage of such a system is that the global address space provides a user-friendly programming perspective to memory and the data sharing between tasks is both fast and uniform due to the proximity of memory to the CPUs. However the primary disadvantage of such a system is the lack of scalability between memory and CPUs. Adding more CPUs can geometrically increase traffic on the shared memory-CPU path, and for the cache coherent systems, geometrically increase traffic associated with cache/memory management. It becomes increasingly difficult and expensive to design and produce shared memory machines with ever increasing numbers of processors.

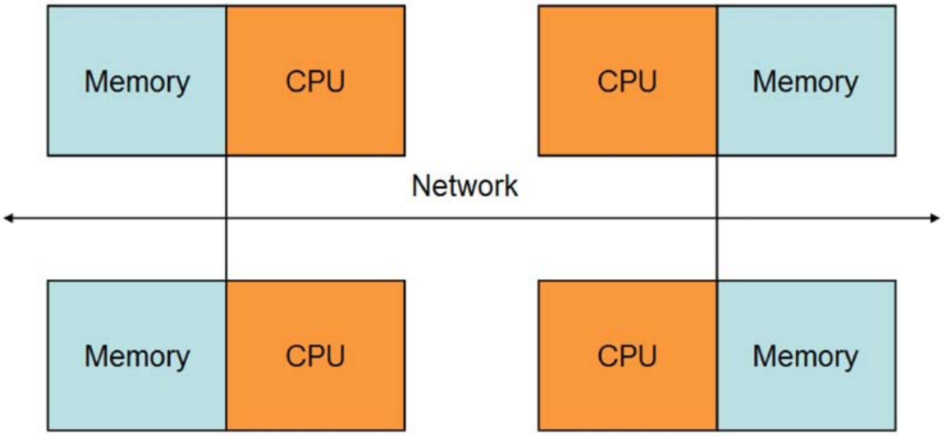


Figure B.4: Illustration of a distributed memory architecture

B.1.2 Distributed memory

In distributed memory machines, each processor has its own independent memory. Memory addresses in one processor do not map to another processor, so there is no concept of global address space across all processors. Distributed memory systems require a communication network to connect inter-processor memory. Because each processor has its own local memory, it operates independently. The changes it makes to its local memory have no effect on the memory of other processors. Hence, the concept of cache coherency does not apply. When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. This is done by passing messages between the processors, typically using a function library such as Message Passing Interface (MPI). Synchronization between tasks is likewise the programmer's responsibility. A schematic diagram of this system is shown in figure (B.4). Simple physical network arrangements such as rings, meshes or the torus, called topologies, are used for direct communication between the processors to avoid excess complexities that would arise if each processor were directly connected to all the other processors. In this system the memory is scalable with number of processors. We can increase the number of processors and the size of memory increases proportionately. Each processor can rapidly access its own memory without interference and without the overheads incurred with trying to maintain cache coherency. These systems are very cost effective and one can use commodity, off-the-shelf processors and networking to build one of these systems. The main disadvantage of this system is that the programmer is responsible for many of the details associated with data communication between processors. It may also be difficult to map existing data structures, based on global memory, to this memory organization.

B.1.3 Hybrid distributed-shared memory

As the name suggests, this type of architecture is a combination of the shared memory and distributed memory systems. The largest and fastest computers in the world today employ both shared and distributed memory architectures. The shared memory component is usually a cache coherent Symmetric Multiprocessor (SMP) machine. SMP machines are identical processors with equal access and access times to memory. Processors on a given SMP can address that machine's memory as global. The distributed memory

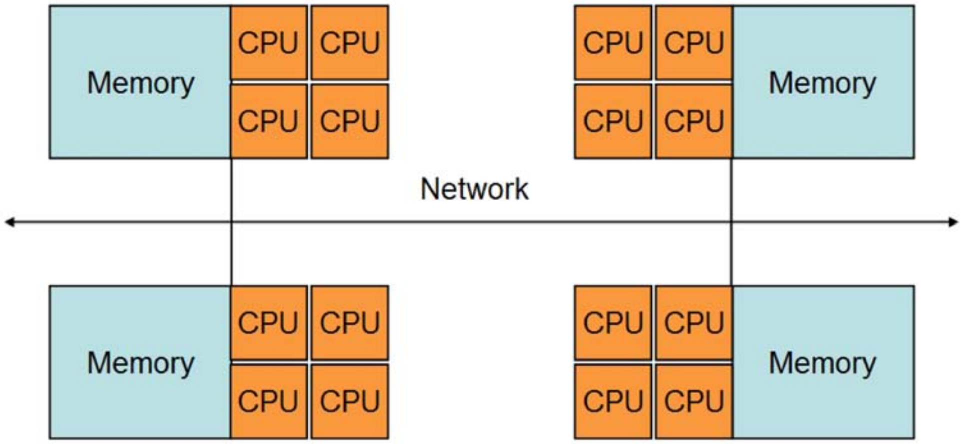


Figure B.5: Illustration of a hybrid memory architecture

component is the networking of multiple SMPs. The SMPs know only about their own memory and not the memory on another SMP. Therefore, network communications are required to move data from one SMP to another just like the distributed systems. A schematic diagram of this system is shown in figure (B.5).

B.2 Designing parallel programs

The first step in developing parallel software is to first understand the problem that you wish to solve in parallel. Before spending time in an attempt to develop a parallel solution for a problem, one should determine whether or not the problem is one that can actually be parallelized. The majority of scientific and technical programs usually accomplish most of their work in a few places in the algorithm and one can start with focusing on these areas of the algorithm to be parallelized. The other important step in parallelizing is to identify the bottlenecks in the algorithm. Its possible that there are areas in the algorithm that are disproportionately slow, or can cause parallelizable work to halt or be deferred, but it may be possible to restructure the program or use a different algorithm to reduce or eliminate unnecessary slow areas.

Programming single-processor system is relatively easy due to single thread execution and single address space. Programming shared memory systems can benefit from the single address space but programming distributed memory systems is difficult due to multiple address spaces and the need to access remote data and communication among the processors. Both the parallel systems (shared memory and distributed memory) offer the ability to perform independent operations on different data (MIMD), to implement task parallelism and at the same time both can be programmed in a data parallel (SIMD) fashion. There are two types of parallelism, namely, data parallelism and task parallelism. Each processor performs the same task on different data in data parallelism whereas each processor performs a different task in task parallelism. Most applications fall somewhere in between these two extremes. The most dominant programming model for shared and distributed memory machines is the Single Program, Multiple Data (SPMD) model. In SPMD model, one source code is written which can have conditional execution based on which processor is executing the copy. All the copies of the code on different processors are started simultaneously and communicate and sync with each other periodically.

Using more processors and more memory in parallel to solve a computational task does not necessarily ensure better performance over the use of a single computer. The performance of parallel computing systems can be diminished due to memory access conflicts, notable in the shared memory machines, or by network conflicts of distributed memory machines. To evaluate the advantage a parallel computing system might have over a sequential machine, objective performance metrics are needed. Some of these measure are known as “speed-up”, “efficiency” and “scalability”.

SpeedUp is the ratio of the time needed for the completion of a sequential algorithm to that time needed for the completion of a parallel algorithm run on P processors. All parallel programs contain some sections that, unfortunately, are serial in nature. These serial sections limit the parallel effectiveness. Amdahl’s Law states this formally and places a strict limit on the speedup (S) that can be realized by using multiple processors.

$$S = \frac{1}{f_s + \frac{f_p}{P}}$$

where f_s is the serial fraction of the code, f_p is the parallel fraction of the code and P is the number of processors.

It usually takes on a small fraction of serial content in a code to degrade the parallel performance. Amdahl’s Law provides a theoretical upper limit on parallel speedup assuming that there is no cost for communication. But, in reality communications will result in further degradation of performance [47].

Concurrent efficiency, can be thought of as speedup per processor. It is defined as the ratio of speedup to the number of processors. Scalability defines the change in efficiency of a parallel algorithm as the number of processors and/or problem size varies. It is commonly based on performance per processor at fixed memory per processor. An important presumption in scalability analysis is that the chosen algorithm and problem size can be carried out on one processor. If a problem is too large for a sequential machine, then an adjusted scalability analysis should be considered. In any case, scalability issues should be examined during the development of parallel code to assure efficiency as the number of processors change.

B.2.1 Partitioning

One of the first steps in designing a parallel program is to break the problem into discrete parts of work that can be distributed to multiple tasks/processors. This is known as decomposition or partitioning.

B.2.1.1 Data distribution

Most of the scientific and engineering problems deal with matrix and vector operations. In data partitioning, the data associated with a problem is decomposed. The matrices and vectors are partitioned such that the operations on the partitioned matrices would be mostly independent. Each parallel task then works on a portion of the data. After the operations on partitioned data are complete the results are shared with the other processors, if needed, in order to perform more operations.

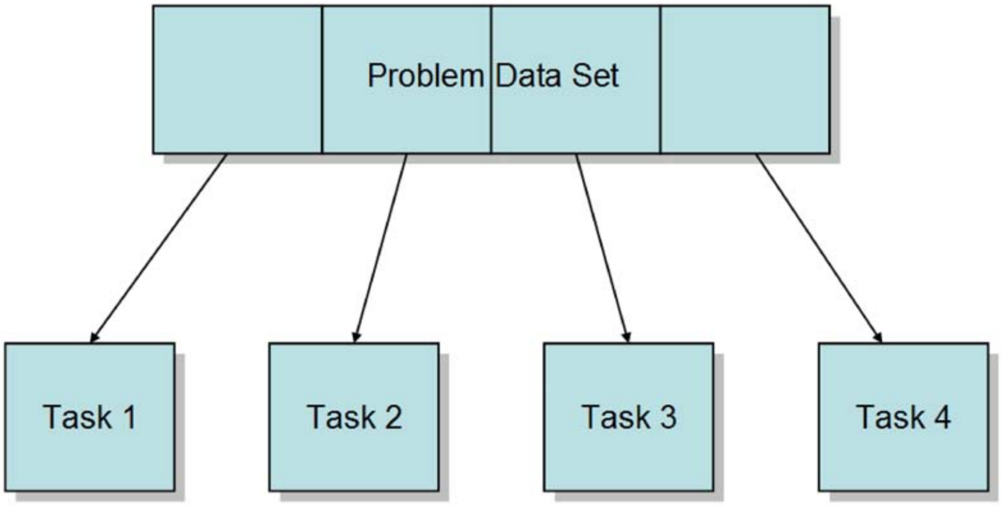


Figure B.6: Illustration of data distribution [37]

There are different ways to partition data. Block distribution and cyclic distribution. Block distribution is where we split the problem such that one contiguous chunk of the data is operated on by a single processor. You can distribute the contiguous data by rows, by columns or by quadrants. These different ways are illustrated in the figure (B.6). The red, blue, yellow and green colors indicate different processors.

The other way to distribute the data is in a cyclic manner. In cyclic distribution, the data is not distributed in contiguous partitions but in non-contiguous partitions as illustrated in the figure (B.7). Cyclic partition can also be distributed by rows, by columns or by quadrants. This type of partition is mostly used in the out-of-core algorithms where the entire data set cannot

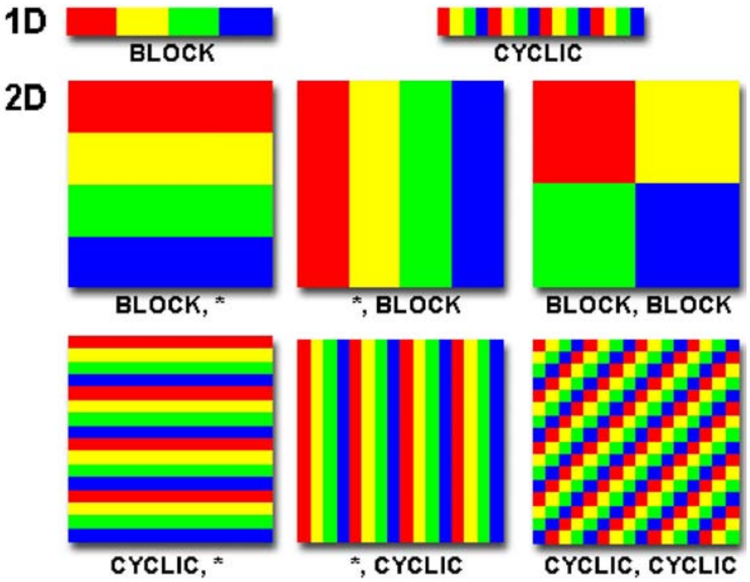


Figure B.7: Illustration of block and cyclic distributions [37]

fit in the total memory available with the processors, in which case the data resides on the physical disk and is partially read into the memory. The type of partition to be used for the data completely depends upon algorithm to be parallelized.

B.2.1.2 Task distribution

In this approach, the focus is on the computation that is to be performed rather than on the data manipulated by the computation. The problem is decomposed according to the work that must be done. Each processor then performs a portion of the overall work. Task distribution lends itself well to problems that can be split into different independent problems and those that can work on the same or different data simultaneously. Figure (B.8) illustrates task distribution.

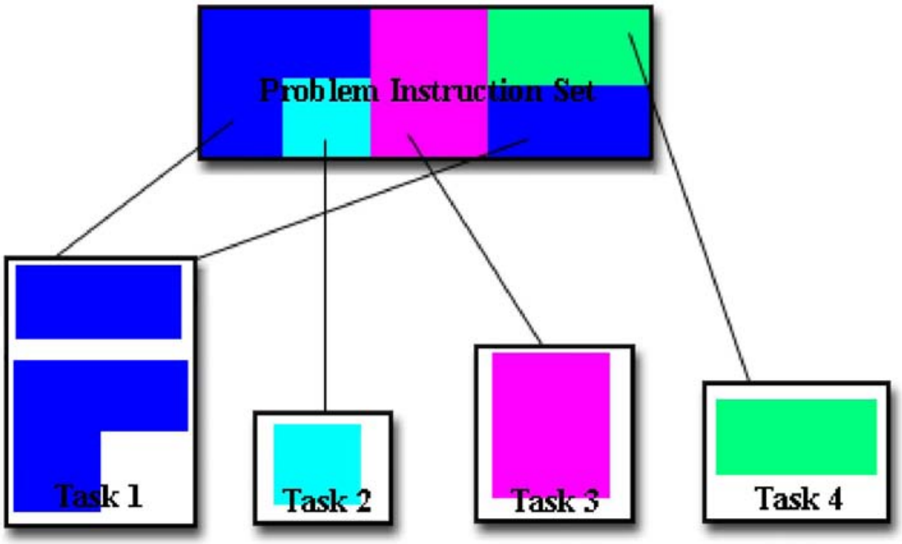


Figure B.8: Illustration of functional distribution [37]

B.2.2 Communications

The need for communications between tasks depends on the problem. Some problems can be distributed to the processor such that a task performed by one processor is completely independent from those performed by the other processor and at no point in the algorithm is there a dependence on the results obtained by some other processor. Such problems will have no need for communication between the processors. Most parallel applications are not quite so simple, and do require tasks to share data with each other.

The two factors to consider when designing the program's inter-task communications are the cost of communication and synchronous vs. asynchronous communications. When communication between two processors is performed, the machine cycles and resources that could be used for computation are instead used to package and transmit data. Communications frequently require some type of synchronization between tasks, which can result in tasks spending time "waiting" instead of performing computations. Also, the competing communication traffic can saturate the available network bandwidth, further aggravating performance problems.

The cost or time, needed to send a message of length n bytes long between two directly connected processors without conflicts, can be modeled as,

$$\alpha + n\beta$$

where α is a fixed startup cost needed to get the attention of the other processors independent of the message size, and β is the transmission cost per unit of data. The term α is referred to as the latency of communication, commonly expressed as microseconds. The reciprocal of β is called the communication bandwidth commonly expressed as megabytes/sec. Sending many small messages can cause latency to dominate communication overheads. It is often more efficient to package small messages into a larger message, thus increasing the effective communications bandwidth.

B.2.3 Synchronization

Synchronous communications require some type of “handshaking” between tasks that are sharing data. This can be explicitly structured in code by the programmer, or it may happen at a lower level unknown to the programmer. Synchronous communications are often referred to as blocking communications since other work must wait until the communications have completed. Asynchronous communications allow tasks to transfer data independently from one another. For example, task 1 can prepare and send a message to task 2, and then immediately begin doing other work, with no dependence on when task 2 actually receives the data. Asynchronous communications are often referred to as non-blocking communications since other work can be done while the communications are taking place. Interleaving computation with communication is the single greatest benefit for using asynchronous communications. There are mainly three type of synchronization used:

- Barrier

  In this case usually all the tasks are involved. Each task performs its work until it reaches the barrier and then stops. When the last task reaches the barrier, all tasks are synchronized and the tasks are automatically released to continue their work. In other cases, after a barrier, often a serial section of work is executed.

- Lock

In the case of lock, number of tasks can be involved. This is typically used to serialize (protect) access to global data or a section of code and only one task at a time may own the lock flag. The first task to acquire the lock "sets" it. This task can then safely (serially) access the protected data or code. The other tasks can attempt to acquire the lock but must wait until the task that owns the lock releases it.

- Synchronous communication operations

  These operations involve only those tasks that are executing a communication operation. When a task performs a communication operation, some form of coordination is required with the other task(s) participating in the communication as discussed in section (B.2.2) For example, before a task can perform a send operation, it must first receive an acknowledgment from the receiving task that it is OK to send.

B.2.4 Data dependencies

A dependence exists between program statements and hence the order of statement execution affects the results of the program. A data dependence results from multiple use of the same location(s) in storage by different tasks/processors. Dependencies are important to parallel programming because they are one of the primary inhibitors to parallelism. In a distributed memory architecture, communication of the required data at synchronization points is required and in a shared memory architecture a synchronization of read/write operations between tasks is necessary to take care of the data dependence.

B.2.5 Load balancing

Load balancing refers to the distribution of tasks in such a way as to insure the most time efficient parallel execution. It can be considered a minimization of task idle time. Load balancing is important to parallel programs for performance reasons. If tasks are not distributed in a balanced way, you may end up waiting for one task to complete while other tasks are idle. For example, if all tasks are subject to a barrier synchronization point, the slowest task will determine the overall performance. To achieve a good load balance we equally partition the work each task receives. When using array or matrix operations where each task performs similar work, data should be distributed among the tasks evenly. For loop iterations, where the work done in each iteration is similar, the iterations should be evenly distributed across the tasks. If a heterogeneous mix of machines with varying performance characteristics are being used, some type of performance analysis tool should be used to detect any load imbalances and work should be adjusted accordingly. Dynamic work assignment should be used in certain classes of problems that result in load imbalance even if the data is distributed evenly among the tasks. An example of such a situation would be the use of a sparse matrix for computation. While some tasks will have actual data to work on, some other tasks will have mostly zeros. There are some classes of problems where the load imbalances occur dynamically within the code, in which case it becomes necessary to design the algorithm to detect and handle the load imbalances.

B.2.6 Granularity

In parallel computing, granularity is a qualitative measure of the ratio of computation to communication. Periods of computation are typically separated from periods of communication by synchronization events.

- Fine-grain Parallelism

  In a fine-grain parallelism, relatively small amount of computational work is done between communication and synchronization events leading to a low computation to communication ratio. This implies high communication overhead and less opportunity for performance enhancement but it facilitates load balancing. If granularity is too fine it is possible that the overhead required for communications and synchronization between tasks takes longer than the computation.

- Coarse-grain Parallelism

  In a coarse-grain parallelism, relatively large amount of computational work is done between communication and synchronization events leading to a high computation to communication ratio. It provides more opportunity for performance increase but makes it difficult to balance load efficiently.

B.3 Message passing interface (MPI)

Communication using message-passing between processors enables parallel computation to take place. MPI (Message-Passing Interface) is a portable standard for programming parallel computers. MPI uses the message-passing paradigm which is well suited for computing on distributed-memory machines. Of course, message-passing can also be used on shared-memory multiprocessors. Compared to other message-passing systems, MPI is more like a superset than a subset, and hence, porting programs to MPI from other message-passing systems is usually relatively straightforward. With MPI, the more complex features are not needed in simple and straightforward parallel programs. MPI is also designed to make efficient implementations possible, so a program using MPI on a certain system should also run relatively fast on another system. MPI is a library and not a language, that is used to support the basic computational model, which is the collection of processors communicating with messages. It is crucial to know which tasks must communicate with each other, during the design stage of a parallel code. MPI involves four basic procedures:

- Initialization and Termination

  All processes must initialize and finalize MPI and must include header files (`mpi.h`) that provide basic definitions and types.

- Setting up the communicators

  Communicators define collections of processes that are allowed to communicate with each other. `MPI_COMM_WORLLD` is a default communicator that encompasses all the processors. Communicators subdivide the nodes into groups, to facilitate performance and clarity of code. Each communicator answers two fundamental questions for the processes; total number of processors and what is the processor id of a particular processor.

- Point-to-point communication

  This type of communication involves sending data from one point (processor/task) to another where one task sends while the other task receives. It is important to realize that all the tasks execute the "same" code.

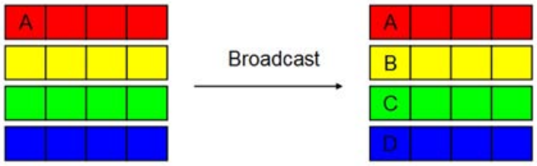
- Collective communication

  This type of communication involves all processors within a communicator. Collective communicators are all blocking and do not use message tags. There are three types of collective communications: synchronization, data movements and computation.

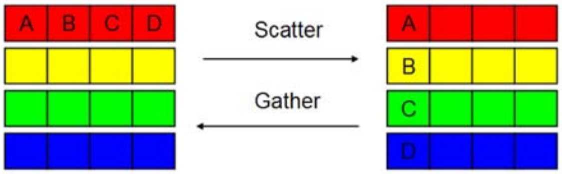
B.4 Some MPI calls

Some of the most commonly used MPI communication calls are described in this section and the figure (B.9) illustrates them. The red, yellow, green and blue colors represent different processors.

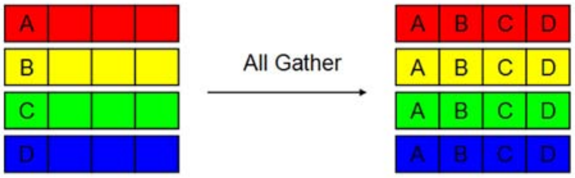
- `MPI_Init() and MPI_Finalize()`



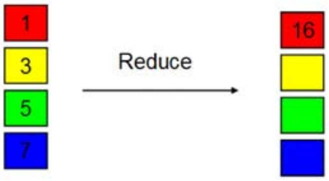
(a) MPI_broadcast



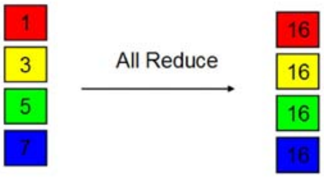
(b) MPI_scatter and MPI_gather



(c) MPI_allgather



(d) MPI_reduce (SUM)



(e) MPI_allreduce (SUM)

Figure B.9: Illustration of MPI communications

These are required for starting and ending execution of an MPI program. Their actions may be implementation dependent. For instance, if the platform is an Ethernet-based cluster , `MPI_Init()` will probably set up the TCP/IP sockets via which the various nodes communicate with each other. On an Infniband-based cluster, connections in the special Infiniband network protocol will be established. On a shared-memory multiprocessor, an implementation of MPI that is tailored to that platform would take very different actions.

- `MPI_Comm_size() and MPI_Comm_rank()`

  `MPI_Comm_size()` call determines how many nodes are participating in the computation within the communicator. `MPI_COMM_WORLD` is the default communicator that defines a group of all the processors. `MPI_Comm_rank()` call determines this nodes ID number, called its rank, within its group. Even though the nodes are all running a copy of the same program, they are typically working on different parts of the programs data. So, the program needs to be able to sense which node it is running on, so as to access the appropriate data.

- `MPI_Send() and MPI_Recv()`

  This is the most simple form of communication between the two processors. `MPI_Send()` sends a buffer of data from one processor to the other and `MPI_Recv()` receives the same buffer from that processor. For the send operation, the usual things to be specified are the data, with starting address and length in bytes for the data type and its destination processor rank. For the receiving processor, the minimal describing arguments will be those which designate where in the local memory the data will be placed, address and length, along with an integer which identifies the origin of the incoming message. Screening parameters which enable a processor to control which message it receives based on data type and origin are necessary to ensure matching of sent and received messages.

- `MPI_Bcast()`

  All nodes participate in a "broadcast" operation, in which a particular node sends a buffer of a particular data type to all the other processors of the communicator groups. Prior to this operation, one processor contains a piece of the data, i.e. a vector and after the broadcast, all the processors contain a copy of the data.

- `MPI_Reduce() and MPI_Allreduce()`

  All nodes in this group participate in a "reduce" operation. `MPI_Reduce()` combines the elements provided in the input buffer of each process in the group, using a defined operation, and returns the combined value in the output buffer of the process with rank root. The type of reduce operation could be "sum", "minimum" etc. Prior to this operation, each processor contains a vector. `MPI_Reduce()` performs the defined reduce operation on all vectors, element wise, and then sends the results to one processor. `MPI_Allreduce()` has the same behavior as `MPI_Reduce()` except that instead of just depositing the result at one node, it does so at all nodes. The illustration in the figures (B.9d) and (B.9e) show the example of the reduce with "sum" operation.

- `MPI_Gather() and MPI_Allgather()`

  A classical approach to parallel computation is to first break the data for the application into chunks, then have each node work on its chunk, and then gather all the processed chunks together at some node. The MPI function `MPI_Gather()` does this. All nodes participate in a "gather" operation, in which each node contributes data, consisting of certain number of MPI data type, from a certain location in its program. All that data is strung together and deposited at the location mind in the program running at node 0.

- `MPI_Scatter() and MPI_Allscatter()`

  `MPI_Scatter()` is the opposite of `MPI_Gather()`, i.e. it breaks long data into chunks which it parcels out to individual nodes. Prior to this, the operation on processor contains a vector of length n elements and after scatter, each processor owns a piece of the vector of length s/P, where P is the number of processor. `MPI_Allscatter()` is equivalent to performing a "scatter" on every processor.

- `MPI_Barrier()`

`MPI_Barrier()` implements a barrier for a given communicator. The name of the communicator is the sole argument for the function. This call is typically used before data/result synchronization between the processors.

Appendix C

Gauss quadrature rules

All of the regularization criteria discussed in this study require the evaluation of matrix moments of the form

$$\begin{aligned} \alpha(\mu) &= b^T(AA^T+\mu I)^p b \\ \nu(\mu) &= b^T A(A^T A+\mu I)^p A^T b \qquad \text{(C.1)} \end{aligned}$$

The exponent p is always a negative integer. If the matrix A is not too large, then direct methods can be used to compute these moments. As the size of the matrix increases, the direct evaluation of these moments becomes less and less feasible. Iterative methods are used in such cases to find the upper bound and lower bound to these moments. The bounds become tighter as the iteration index k increases. This chapter provides a summary of the derivation of the bounds on the moments in the equations (C.1) and the reader is encouraged to read [5] and [19], for more details.

We introduce the function

$$\phi_\mu(t) := (t+\mu)^{-p} \qquad \text{(C.2)}$$

Thus the equations (C.1) can be repesented in terms of the function $\phi_\mu(t)$ as,

$$\begin{aligned} \alpha(\mu) &= b^T \phi_\mu(AA^T) b \\ \nu(\mu) &= (A^T b)^T \phi_\mu(A^T A)(A^T b) \end{aligned} \tag{C.3}$$

The key to obtaining computable bounds for $\alpha(\mu)$ and $\nu(\mu)$ is to express these quantities as Stieltjes integrals. Assume for each notation that $m \geq n$ and introduce the singular value decomposition (SVD) of the matrix A,

$$A = \tilde{U} \begin{bmatrix} \tilde{\Sigma} \\ 0 \end{bmatrix} \tilde{V}^T \tag{C.4}$$

where $\tilde{U} \in \Re^{m \times m}$ and $\tilde{\Sigma}, \tilde{V} \in \Re^{n \times n}$, with $\tilde{U}$ and $\tilde{V}$ orthogonal, and $\tilde{\Sigma} = diag(\tilde{\sigma_1}, \tilde{\sigma_2}, \cdots, \tilde{\sigma_n})$. Then

$$\begin{aligned} AA^T &= \tilde{U} \Lambda \tilde{U}^T \\ A^T A &= \tilde{V} \hat{\Lambda} \tilde{V}^T \end{aligned} \tag{C.5}$$

where,

$$\begin{aligned} \Lambda &:= diag(\lambda_1, \lambda_2, \cdots, \lambda_n, 0, 0, \cdots, 0) \in \Re^{m \times m} \\ \hat{\Lambda} &:= diag(\lambda_1, \lambda_2, \cdots, \lambda_n) \in \Re^{n \times n} \end{aligned}$$

and

$$\lambda_i = \tilde{\sigma}_i^2 \quad \text{for} \quad 1 \leq i \leq n.$$

Now let,

$$\begin{aligned} h = [h_1, h_2, \cdots, h_m]^T &:= \tilde{U}^T b \\ \hat{h} = [\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_n]^T &:= \tilde{V}^T A^T b \end{aligned}$$

and substitute the SVD of A (C.4) into (C.1), we obtain

$$\alpha(\mu) = h^T \phi_\mu(\Lambda) h = \sum_{k=1}^{n} \phi_\mu(\lambda_k) h_k^2 + \phi_\mu(0) \sum_{k=n+1}^{m} h_k^2 = \int_{-\infty}^{\infty} \phi_\mu(t) d\omega(t) \quad \text{(C.6)}$$

$$\nu(\mu) = \hat{h}^T \phi_\mu(\hat{\Lambda}) \hat{h} = \sum_{k=1}^{n} \phi_\mu(\lambda_k) \hat{h}_k^2 = \int_{-\infty}^{\infty} \phi_\mu(t) d\hat{\omega}(t) \quad \text{(C.7)}$$

The distribution functions ω and $\hat{\omega}$, defined by equations (C.6) and (C.7), respectively, are non-decreasing step functions with jump discontinuities at the eigenvalues λ_k. Moreover, the finction ω generally has a jump discontinuity at the origin when $m > n$.

We can now show how Gauss and Gauss-Radau quadrature rules can be applied to compute the upper and lower bounds of the quantities $\alpha(\mu)$ and $\nu(\mu)$ using their representations (C.6) and (C.7), respectively, in terms of Stieltjes integrals. Define the inner product induced by the distribution function $\hat{\omega}$ introduced in equation (C.7),

$$\langle f, g \rangle := \int_{-\infty}^{\infty} f(t) g(t) d\hat{\omega}(t) = \sum_{k=1}^{n} f(\lambda_k) g(\lambda_k) \hat{h}_k^2 = \hat{h}^T f(\hat{\Lambda}) g(\hat{\Lambda}) \hat{h} \quad \text{(C.8)}$$

and let $\hat{q}_1, \hat{q}_2, \hat{q}_3, \ldots$ be the family of orthonormal polynomials with positive leading coefficients with respect to this inner product; thus,

$$\langle \hat{q}_k, \hat{q}_j \rangle = \begin{cases} 0, & \text{if } k \neq j \\ 1, & \text{if } k = j \end{cases} \quad \text{(C.9)}$$

The polynomials, $\hat{q}_k$ satisfy a three-term recurrence relation of the form

$$t\hat{q}_{k-1}(t) = \hat{\beta}_k \hat{q}_k(t) + \hat{\alpha}_k \hat{q}_{k-1}(t) + \hat{\beta}_{k-1} \hat{q}_{k-2}(t), \qquad k = 1, 2, 3, \cdots \quad \text{(C.10)}$$

where $\hat{q}_{-1}(t) := 0$, $\hat{q}_0(t) := \langle 1,1\rangle^{-1/2}$ and $\hat{\beta}_0 := 0$. Since the distribution function $\hat{\omega}$ has at most n distinct points of increase, there are at most n orthonormal polynomials $\hat{q}_j$.

The k steps of Lanczos algorithm applied to a the matrix A^TA with initial vector A^Tb yields the symmetric positive definite or semidefinite tridiagonal matrix

$$\hat{T}_k := \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & & \\ \hat{\beta}_1 & \hat{\alpha}_2 & \hat{\beta}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \hat{\beta}_{k-2} & \hat{\alpha}_{k-1} & \hat{\beta}_{k-1} \\ & & & \hat{\beta}_{k-1} & \hat{\alpha}_k \end{bmatrix} \tag{C.11}$$

whose entries are the first $2k-1$ coefficients in the recurrence relation (C.10). The k-point Gauss quadrature rule associated with the distribution function $\hat{\omega}$ can be expressed in terms of the tridiagonal matrix $\hat{T}_k$ as follows,

$$\hat{\mathcal{G}}_k(f) = \|A^Tb\|^2 e_1^T f(\hat{T}_k) e_1 \tag{C.12}$$

where, e_j denotes the j^{th} axis vector.

Introduce the Choleskey factor of $\hat{T}_k$,

$$\hat{T}_k := \begin{bmatrix} \hat{\gamma}_1 & & & & \\ \hat{\delta}_1 & \hat{\gamma}_2 & & & \\ & \ddots & \ddots & & \\ & & \hat{\delta}_{k-2} & \hat{\gamma}_{k-1} & \\ & & & \hat{\delta}_{k-1} & \hat{\gamma}_k \end{bmatrix} \tag{C.13}$$

as well as the matrix

$$\bar{\hat{C}}_{k-1} = \begin{bmatrix} \hat{C}_{k-1} \\ \hat{\delta}_k e_{k-1}^T \end{bmatrix} \in \Re^{k\times(k-1)} \tag{C.14}$$

which is made up of the first $k-1$ columns of $\hat{C}_k$.

We use the matrix $\bar{\hat{C}}_{k-1}$ in a representation analogous to C.12 of the k-point Gauss-Radau quadrature rule associated with the distribution function $\hat{\omega}(t)$ with one node assigned to the origin,

$$\hat{\mathcal{R}}_k(f) = \|A^T b\|^2 e_1^T f(\bar{\hat{C}}_{k-1}\bar{\hat{C}}_{k-1}^T)e_1 \tag{C.15}$$

where, e_j denotes the j^{th} axis vector. Thus the bounds on $\nu(\mu)$ as derived by (ref??) is

$$\hat{\mathcal{G}}_k(\phi_\mu) \le \nu(\mu) \le \hat{\mathcal{R}}_k(\phi_\mu) \tag{C.16}$$

When computing the bounds for ν_μ defined by equation (C.1), let $T_k \in \Re^{l\times l}$ denote the tridiagonal matrix obtained by the k steps of the Lanczos process applied to AA^T with initial vector b, and let C_k denote the lower Cholesky factor of T_k. Then analogous to the representation C.12, the k-point Gauss quadrature rule, with respect to the function $\omega(t)$ defined by C.6, can be written as

$$\mathcal{G}_k(f) = \|b\|^2 e_1^T f(T_k)e_1 \tag{C.17}$$

where, e_j denotes the j^{th} axis vector.

Similarly to C.15, the k-point Gauss-radau rule, associated with the distribution function $\omega(t)$ with one node assigned to the origin,

$$\mathcal{R}_k(f) = \|A^T b\|^2 e_1^T f(\bar{C}_{k-1}\bar{C}_{k-1}^T)e_1 \tag{C.18}$$

where, e_j denotes the j^{th} axis vector and the matrix $\bar{C}_{k-1}$ consists of $k-1$ first columns of the matrix C_k and the bounds on $\alpha(\mu)$ as derived by (ref??) is

$$\mathcal{G}_k(\phi_\mu) \leq \nu(\mu) \leq \mathcal{R}_{k+1}(\phi_\mu) \tag{C.19}$$

Instead of using the Lanczos tridiagonal matrix, T_k, that is obtained by applying the k-steps of Lanczos process to AA^T and the initial vector b, and then computing its Cholesky factors ($T_k = C_k C_k^T$), we can directly compute the its Cholesky factor by using the k-steps Lanczos bidiagonalization algorithm. It is also shown by (ref??) that the lower bidiagonal Cholesky factor $\hat{C}_k$ of $\hat{T}_k$ as defined in C.11, can be computed without forming $\hat{T}_k$. It is computed using the QR-factorization, $\bar{C}_k = \bar{Q}_k \hat{C}_k^T$, where $\bar{Q}_k \in \Re^{(k+1)\times k}$ satisfies $\bar{Q}_k^T \bar{Q}_k = I_k$ and $\hat{C}_k^T$ is upper bidiagonal.

Bibliography

[1] S. V. Bettadpur and R. J. Eanes. Geographical representation of radial orbit perturbations due to ocean tides: Implications for satellite altimetry. *Journal of Geophysical Research*, 99(C12):24883–24894, 1994.

[2] S. V. Bettadpur, B. E. Schutz, and J. B. Lundberg. Spherical harmonic synthesis and least squares computations in satellite gravity gradiometry. *Journal of Geodesy*, 66(3):261–271, 1992.

[3] G. H. Born and B. D. Tapley. Sequential estimation of the state and the observation-error convariance matrix(Algorithm for simultaneous estimate of spacecraft state and covariance matrix with observation error vector). *AIAA Journal*, 9:212–217, 1971.

[4] D. Calvetti, G. H. Golub, and L. Reichel. Estimation of the L-curve via Lanczos bidiagonalization. *BIT*, 39.

[5] D. Calvetti, P. C. Hansen, and L. Reichel. L-Curve Curvature Bounds via Lanczos Bidiagonalization. *Electronic Transactions on Numerical Analysis*, 14:134–149, 2002.

[6] D. Calvetti, L. Reichel, and A. Shuibi. L-Curve and Curvature Bounds for Tikhonov Regularization. *Numerical Algorithms*, 35(2):301–314, 2004.

[7] J. L. Chen, C. R. Wilson, and K. W. Seo. Optimized smoothing of Gravity Recovery and Climate Experiment (GRACE) time-variable gravity observations. *J. Geophys. Res*, 111, 2006.

[8] J. L. Chen, C. R. Wilson, B. D. Tapley, and S. Grand. GRACE detects coseismic and postseismic deformation from the Sumatra-Andaman earthquake. *Geophys. Res. Lett*, 34(13), 2007.

[9] J. Chung, J. G. Nagy, and D. P. OLeary. A weighted-GCV method for Lanczos-hybrid regularization. *Electronic Transactions on Numerical Analysis*, 28:149–167, 2008.

[10] National Research Council. *Satellite Gravity and the Geosphere: Contributions to the Study of the Solid Earth and Its Fluid Envelopes.* The National Academies Press, 1997.

[11] J. L. Davis, M. E. Tamisiea, P. Elósegui, J. X. Mitrovica, and E. M. Hill. A statistical filtering approach for Gravity Recovery and Climate Experiment (GRACE) gravity data. *Journal of Geophysical Research*, 113 (B4), 2008.

[12] P. Ditmar, J. Kusche, and R. Klees. Computation of spherical harmonic coefficients from gravity gradiometry data to be acquired by the GOCE satellite: regularization issues. *Journal of Geodesy*, 77(7):465–477, 2003.

[13] P. Döll, J. Alcamo, T. Henrichs, F. Kaspar, B. Lehner, T. Rösch, and

S. Siebert. The global integrated water model WaterGAP 2.1. *EuroWasser, Kassel World Water Series*, 5, 2001.

[14] C. Dunn, W. Bertiger, Y. Bar-Sever, S. Desai, B. Haines, D. Kuang, G. Franklin, I. Harris, G. Kruizinga, T. Meehan, et al. Instrument of grace; gps augments gravity measurements. *GPS World*, 14:pp 16–28, February 2003.

[15] R. J. Eanes. Personal communication, 2008.

[16] L. Elden. A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems. *BIT(Nordisk Tidskrift for Informationsbehandling)*, 24(4):467–472, 1984.

[17] M. J. Flynn. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, 21(9):948–960, 1972.

[18] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[19] G. H. Golub and U. von Matt. Tikhonov Regularization for Large Scale Problems. In *Scientific Computing: Proceedings of the Workshop, 10-12 March 1997, Hong Kong.* Springer, 1998.

[20] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2): 215–223, 1979.

[21] B. C. Gunter. Parallel least squares analysis of simulated grace data. Master's thesis, University of Texas at Austin, 2000.

[22] B. C. Gunter. *Computational methods and processing strategies for estimating Earths gravity field*. PhD thesis, The University of Texas at Austin, 2004.

[23] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial Mathematics, 1998.

[24] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial Mathematics, 1998.

[25] P. C. Hansen. *The L-curve and its use in the numerical treatment of inverse problems*. IMM, Department of Mathematical Modelling, Technical University of Denmark.

[26] M. B. Hinga. *Using parallel computation to apply the singular value decomposition (SVD) in solving for large Earth gravity fields based on satellite data*. PhD thesis, The University of Texas at Austin, 2004.

[27] K. H. Ilk, J. Kusche, and S. Rudolph. A contribution to data combination in ill-posed downward continuation problems. *Journal of Geodynamics*, 33(1-2):75–99, 2002.

[28] T. Katagiri. Performance Evaluation of Parallel Gram-Schmidt Re-orthogonalization Methods. *Lecture notes in Compute Science*, pages 302–314, 2003.

[29] W. Kaula. *Theory of Satellite Geodesy.* Blaisdell Publishing Co., Waltham, MA, 1966.

[30] T. P. Keating, P. Taylor, W. Kahn, and F. Lerch. Geopotential research mission, science engineering, and program summary. TM- 86240, NASA, 1986.

[31] M. E. Kilmer and D. P. OLeary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM J. Matrix Anal. Appl*, 22 (4):1204–1221, 2001.

[32] J. R. Kim. *Simulation study of a low-low satellite-to-satellite tracking mission.* PhD thesis, The University of Texas at Austin, 2000.

[33] J. Kusche and R. Klees. Regularization of gravity field estimation from satellite gravity gradients. *Journal of Geodesy*, 76(6):359–368, 2002.

[34] M. L. Larsen and P. C. Hansen. Parallel implementation of lanczos bidiagonalization on convex exemplar. Technical report, 1998.

[35] R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. 1998.

[36] C. L. Lawson and R. J. Hanson. Solving least squares problems. *Prentice-Hall Series in Automatic Computation, Englewood Cliffs: Prentice-Hall*, 1974.

[37] LLNL. Introduction to parallel computing. Website. `https://computing.llnl.gov/tutorials/parallel_comp`.

[38] J. I. MacArthur and A. S. Posner. Satellite-to-satellite range-rate measurement. *IEEE Transactions on Geoscience and Remote Sensing*, Vol GE-23:pp 517–523, July 1985.

[39] T. Mayer-Gurr. ITG-Grace03s: The latest GRACE gravity field solution computed in Bonn Joint Int. In *GSTM and DFG SPP Symposium, Potsdam*, 15-17 Oct .

[40] R. D. Ray and S. B. Luthcke. Tide model errors and GRACE gravimetry: towards a more realistic assessment. *Geophysical Journal International*, 167(3):1055–1059, 2006.

[41] T. Regińska. A Regularization Parameter in Discrete Ill-Posed Problems. *SIAM Journal on Scientific Computing*, 17:740, 1996.

[42] H. J. Rim. *TOPEX orbit determination using GPS tracking system*. PhD thesis, The University of Texas at Austin, 1992.

[43] R. Sanso and R. (Eds.) Rummel. *Theory of Satellite Geodesy and Gravity Field Determination, vol. 25 of Lecture Notes in Earth Sciences*. Springer-Verlag, Berlin, 1989.

[44] E. J. O. Schrama, B. Wouters, and D. A. Lavallée. Signal and noise in Gravity Recovery and Climate Experiment (GRACE) observed surface mass variations. *Journal of Geophysical Research*, 112(B8), 2007.

[45] S. Swenson and J. Wahr. Methods for inferring regional surface-mass anomalies from Gravity Recovery and Climate Experiment (GRACE) measurements of time-variable gravity. *Journal of Geophysical Research (Solid Earth)*, 107(B9), 2002.

[46] S. Swenson and J. Wahr. Post-processing removal of correlated errors in GRACE data. *Geophys. Res. Lett*, 33, 2006.

[47] TACC. *High Performance Computing - Training Manual.* Jan 2006.

[48] B. Tapley, D. Yuan, and C. Shum. Gravity field determination and error assessment techniques. In D. Forsyth, editor, *1988 Chapman Conference on Progress in the Determination of the Earths Gravity Field*, Sept 1988.

[49] B. D. Tapley. Statistical Orbit Determination Theory. *Recent Advances in Dynamical Astronomy*, 39:396425, 1973.

[50] B. D. Tapley, S. Bettadpur, M. Watkins, and C. Reigber. The gravity recovery and climate experiment: Mission overview and early results. *Geophys. Res. Lett*, 31(9):L09607, 2004.

[51] B. D. Tapley, B. E. Schutz, and G. H. Born. *Statistical Orbit Determination.* Academic Press, 2004.

[52] J. B. Thomas. An Analysis of Gravity-Field Estimation Based on Intersatellite Dual-1-Way Biased Ranging, 1999.

[53] P. Touboul, B. Foulon, M. Rodrigues, and J. P. Marque. In orbit nano-g measurements, lessons for future space missions. *Aerospace Science and Technology*, 8(5):431–441, 2004.

[54] Universität Bonn. ITG-Grace03 Gravity Field Model. Website. `http://www.geod.uni-bonn.de/itg-grace03.html`.

[55] D. A. Vallado. *Fundamentals of Astrodynamics and Application.* The McGraw Hill Companies, Inc, 1997.

[56] J. Vetter, R. S. Nerem, P. Cefola, and H. Hagar. A historical survey of earth gravitational models used in astrodynamics from sputnik and TRANSIT to GPS and TOPEX. *AAS/AIAA Astrodynamics Specialist Conference*, AAS 93-620:16–19, 1993.

[57] J. Wahr, M. Molenaar, and F. Bryan. Time variability of the Earth's gravity field: Hydrological and oceanic effects and their possible detection using GRACE. *Journal of Geophysical Research*, 103:30–30, 1998.

[58] D. S. Watkins. *Fundamentals of matrix computations.* John Wiley & Sons, Inc. New York, NY, USA, 2002.

[59] M. Wolf. Direct measurements of the earth's gravitational potential using a satellite pair. *Journal Of Geophysical Research*, Vol 74:pp 5295–5300, 1969.

[60] B. Wouters and E. J. O. Schrama. Improved accuracy of GRACE gravity solutions through empirical orthogonal function filtering of spherical harmonics. *Geophysical Research Letters*, 34(23), 2007.

[61] S. C. Wu, G. Kruizinga, and W. Bertiger. Algorithm Theoretical Basis Document for GRACE Level-1B Data Processing V1. 2.

[62] D. N. Yuan. *The Determination and Error Assessment of the Earths Gravity Field Model*. PhD thesis, The University of Texas at Austin, 1991.

Vita

Himanshu Vijay Save was born in Mumbai, India on 17 September 1977, the son of Vijay Save and Hemalata Save. He received the Bachelor of Technology degree in Engineering from the Indian Institute of Technology (IIT), Bombay in 2000. He was accepted to graduate studies program in Aerospace Engineering at the University of Texas at Austin and is working with the Gravity Recovery and Climate Experiment (GRACE) mission group at The Center for Space Research (CSR) at the university as a Graduate Research Assistant.

Permanent address: 12402 Bedrock Trail
Austin, Texas 78727

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.